

Entwicklung von vollautomatischem Deployment von Microsoft Betriebssystemen im virtuellen Umfeld

Development of Fully Automatic Deployment of Microsoft Operating Systems in a Virtual Environment

Helmut Thomas

Diplomarbeit

Betreuer: Prof. Dr. Rainer Oechsle

Koblenz, 29.02.2008

Vorwort

Die Idee zu vorliegender Arbeit ist im Rahmen des Projektes "Migration von Windows NT zu Windows XP bzw. Windows Server 2003" entstanden. Ziel war Verbesserung und Vereinfachung von Client- und Serverinstallationen. Auch wenn die Migration abgeschlossen ist, müssen immer wieder Clients und Server ausgetauscht werden. Daher werden das daraus gewonnene Konzept und die Ergebnisse auch nach Abschluss der Arbeit Fortbestand haben und weiter entwickelt werden.

Danken möchte ich Herrn Prof. Dr. Oechsle, der meine Arbeit betreut hat und mich stets in dem Gedanken bestärkt hat, die Diplomarbeit in der vorliegenden Art und Weise umzusetzen.

Besonderer Dank gilt meiner Freundin Silke Billker (Miss "Duden") für ihre Geduld und Unterstützung während meiner Arbeit und insbesondere dafür, dass sie sich die Zeit nahm, dieses Dokument auf sprachliche Schwachstellen und Tippfehler zu untersuchen.

Gebührender und großer Dank geht gleichfalls an meinen Kollegen und direkten Stellvertreter Matthias Kray für seine konstruktive Kritik, nachdem er meine Arbeit gelesen hat, und dafür, dass er mich während meiner Abwesenheit immer klaglos vertreten hat.

Mein weiterer Dank gilt Christoph Link und Geschäftsführer Oliver Büring von der Firma Neton, die es mir ermöglicht haben, die Büroräume in Koblenz für die Dauer meiner Diplomarbeit zu nutzen. Ebenfalls danken möchte ich der Kreisverwaltung Mayen-Koblenz für ihre langjährige Unterstützung des berufsbegleitenden Fernstudiums und der Möglichkeit, meine zahlreichen Überstunden so flexibel und unkompliziert abzubauen.

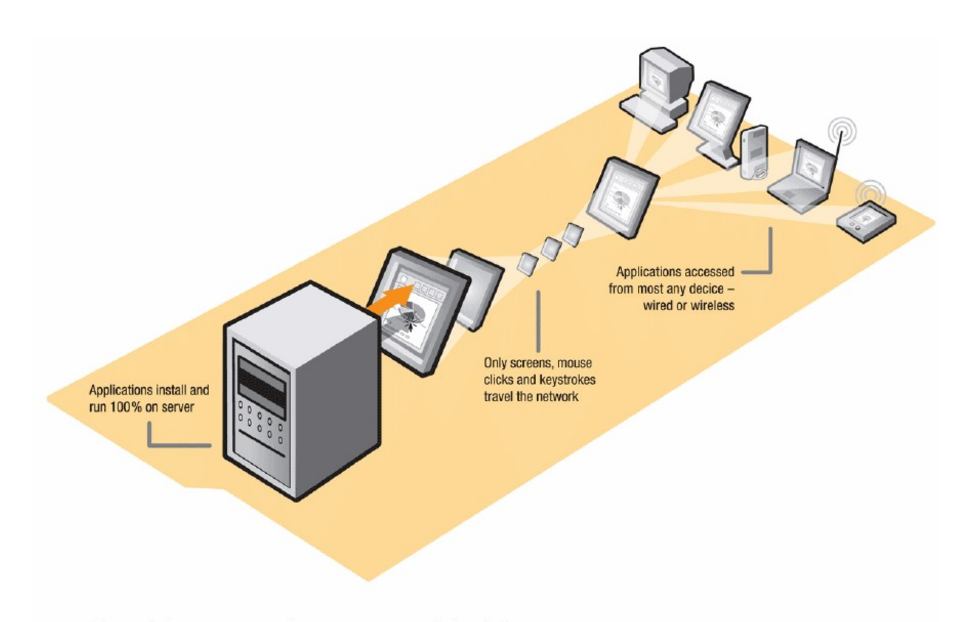
Kurzfassung

Die vorliegende Diplomarbeit befasst sich mit der vollautomatischen Verteilung von Betriebssystemen im Microsoft-Active-Directory-Umfeld mit Integration in eine Terminalserver-Umgebung, die durch Citrix Metaframe bzw. dem Nachfolgeprodukt Citrix Presentation Server geprägt ist. Dabei wird von einer Infrastruktur mit 75% Thin-Clients und 25% Fat-Clients ausgegangen. Fat-Clients werden mit einem Microsoft- und Thin-Clients mit einem Linux-Betriebssystem ausgestattet. Im Serverbereich werden die Maschinen mit MS Windows Server 2003 ausgestattet.

Der Verzeichnisdienst Microsoft Active Directory bietet die Möglichkeit Objekte (Benutzer, Gruppen, Computer, Gruppenrichtlinien, etc.) hierarchisch strukturiert abzulegen. Eine Abfrage dieser Informationen mittels TCP/IP über das Protokoll LDAP (Lightweight Directory Access Protocol) ist nach vorheriger Authentifizierung möglich.

Im Bereich des Server Based Computing (SBC) ist Citrix Presentation Server eine Software, die auf den Windows Terminal Diensten aufsetzt und diese Funktionen noch erweitert und verfeinert (CPU-Auslastungsmanagement, Bandbreitenbegrenzung, erweiterte Lastverteilung, Ressourcenmanagement, universeller Druckertreiber, etc.) [CTX07]. *"Ein Terminal ist ein Datensichtgerät mit angeschlossener Tastatur, das alle Tastendrücke des Benutzers über eine Netzwerkverbindung an einen zentralen Großrechner sendet... Der wesentlichste Unterschied zwischen den Terminals und dem SBC ist der, dass der zentrale Rechner kein Mainframe mehr ist, sondern ein leistungsstarker Server auf Basis der x86 Architektur, der unter Microsoft Windows betrieben wird. Und die Nutzung der durch diesen Server angebotenen Rechenleistung ist entweder für eine spezielle Software oder ein dediziertes Endgerät, den sogenannten Thin Client möglich."* [FhG05]

Ein Fat-Client verarbeitet Daten lokal (im Gegensatz zu einem Thin-Client) und bedient sich dabei in der Regel einer grafischen Oberfläche. Fat-Clients sind vollwertig ausgestattete, leistungsfähige Desktop-Computer mit lokalen Speichermedien, wie Festplatten, Disketten- und CD/DVD-Laufwerken sowie leistungsstarken Grafikkarten. Fat-Clients liefern einen Großteil der Prozessorleistung in Client-Server-Anwendungen. Ein reiner Thin-Client dagegen ist ein Window-Terminal, das für Remote-Anwendungen ausgelegt ist und aus mindestens Tastatur, Maus, grafischer Ausgabe und Sound besteht. In Client-Server-Anwendungen begnügt sich der Thin-Client mit der Darstellung des Bildschirminhaltes, sowie der Bedienung der E/A-Geräte Tastatur und Maus. Die Prozessorleistung für die Verarbeitung der Anwendung liegt ausschließlich beim Server.



Quelle: Citrix Systems (http://www.citrix.de)

Es werden die Vorteile einer Standardisierung im Hardware- als auch im Softwarebereich verdeutlicht. Weiterhin wird ein Lösungsansatz aufgezeigt, um auch bei inhomogener Hardware von automatischer Betriebssystemverteilung:- zu profitieren.
Das Ergebnis der Diplomarbeit zeigt, wie mit betriebssystemeigenen Bordmitteln und geringem zusätzlichem Software- und Programmieraufwand eine schnelle, vollautomatische und flexible Verteilung der Betriebssysteme samt Anwendungssoftware möglich ist. Der Schwerpunkt liegt dabei auf der Ausbringung von Windows XP und der automatischen Integration in den Verzeichnisdienst Active Directory sowie der Installation von MS Windows Server 2003 und anschließender automatischer Installation von Citrix Presentation Server und Integration in eine bestehende Terminalserverumgebung.

Die Aufteilung 75/25 Thin/Fat-Clients ergibt sich aus der Notwendigkeit, ca. 100 verschiedene Fachverfahren abzudecken, die unterschiedliche Hard- und/oder Softwareanforderungen stellen. Leider können diese Anforderungen derzeit nicht ausschließlich mit Thin-Clients erfüllt werden, so dass sich der Anteil von ca. 25% Fat-Clients ergibt.
Ein weiterer Schwerpunkt zeigt die Besonderheiten auf, die im virtuellen Umfeld zu beachten sind. Dazu wird gezeigt, wie man aus einer Echtumgebung eine virtuelle Testumgebung aufbaut und dort das Deployment vornimmt.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ACPI	Advanced Configuration Power Interface
APCI	Advanced Programmable Interrupt Controller
AS/400	Application System 400
BINL	Boot Information Negotiation Layer
CIW	Client Installation Wizard
COLS	Commercial Linux Software
DB	Datenbank
DC	Domänencontroller (engl. Domain Controler)
DHCP	Dynamic Host Configuration Protocol
eLux NG	Embedded Linux für Thin Clients Next Generation
ext3	Third Extended Filesystem
FS	Filesystem (Dateisystem)
FSC	Fujitsu Siemens Computer
ICA	Independent Computing Architecture
IMA	Independent Management Architecture
KVMYK	Kreisverwaltung Mayen-Koblenz
LDAP	Lightweight Directory Access Protocol
MAC	Media Access Control
NBNS	NetBIOS Name Service
NetBIOS	Network Basic Input Output System
NTFS	New Technology File System
NAT	Network Adress Translation
OU	Organisationseinheit (Organizational Unit)
OSS	Open Source Software
PXE	Preboot eXecution Environment
RIS	Microsoft Remote Installation Services
Scout NG	Name der Verwaltungssoftware für eLux NG (übersetzt "Pfadfinder")
SBC	Server Based Computing
SID	Security Identifier
SMB	Server Message Block (auch Lan-Manager oder NetBIOS-Protokoll)
SNMP	Simple Network Management Protocol
TFTP	Trivial File Transfer Protocol
UUID	Universally Unique Identifier
WINS	Windows Internet Naming Service

1 Einleitung

Die Kreisverwaltung Mayen-Koblenz setzt seit 1999 im Serverbereich Microsoft Windows NT Terminalserver mit Citrix Metaframe 1.8 und als Client-Betriebssystem Microsoft Windows NT Workstation ein. Bis ins Jahr 2007 wurde das eingesetzte Betriebssystem auf Microsoft Windows Server 2003 inklusive Verzeichnisdienst Microsoft Active Directory und Citrix Presentation Server 4.0 migriert. Sofern es sich bei den Clients um Fat-Clients handelt, kommt dort Microsoft Windows XP Professional zum Einsatz. Thin-Clients werden mit dem Linux-Betriebssystem eLux NG[1] der Fa. Unicon betrieben. Im Vorfeld wurde eine ausführliche Untersuchung durchgeführt, die zum Ergebnis kam, dass das derzeitige Konzept des Server Based Computing auch zukünftig beibehalten wird. Es wurde dabei das Ziel gesetzt, den Anteil an Thin-Clients von 55% auf 75% zu steigern und den Anteil Fat-Clients entsprechend zu verringern. Das Software-Deployment der Clients (Thin- und Fat-Client, sowie Server) sollte so weit wie möglich automatisiert geschehen.

Die Clients waren nicht in einen Verzeichnisdienst integriert, sondern in einer Windows NT-Domänenstruktur organisiert. In einer NT-Domänenstruktur sind alle Objekte (Benutzer und Computer) in einer flachen, nicht weiter strukturierbaren Organisation angeordnet, so dass alle Objekte in einer Ebene liegen. In dem Verzeichnisdienst Microsoft Active Directory ist dagegen eine organisatorische und strukturelle Gliederung von Benutzern und Computerkonten möglich. Dies wird dadurch erreicht, dass sogenannte Organisationseinheiten (OU[2]) gebildet werden können, in die Benutzer und/oder Computer aufgenommen werden. Dadurch ist zum einen eine bessere Übersichtlichkeit gewährleistet und zum anderen ist es dadurch möglich, einzelnen OUs unterschiedliche Richtlinien zuzuordnen. Man spricht in diesem Zusammenhang auch von Gruppenrichtlinien, da diese - optimal eingesetzt - Gruppen (Benutzergruppen oder OUs) und nicht einzelnen Objekten (Benutzer, Computer) zugeordnet werden, gleichwohl dies natürlich möglich ist. Richtlinien dienen dazu, bestimmte Einstellungen vorzugeben, wie beispielsweise Passwortlänge und dessen Komplexität oder die Proxyeinstellung des Browsers. Ein weiteres wichtiges Merkmal ist die Vererbung, also das Vererben von Eigenschaften durch die gesamte Hierarchie des Verzeichnisdienstebaums.

Die Umstellung der Endgeräte erfolgte im laufenden Betrieb. Die Störung im Dienstbetrieb sollte so gering wie möglich gehalten werden. Die Umstellung betraf sowohl den Austausch der Hardware als auch das Aktualisieren der Softwarekomponenten. Die Aktualisierung der Server erfolgte ebenfalls im laufenden Betrieb, allerdings wird hier eine neue zusätzliche Hardwareplattform parallel zur alten Bestehenden aufgebaut, so dass hier ein nahtloser Übergang stattfindet.

---

[1] *http://www.myelux.de/*

[2] Im Original heißt es *organizational unit* und es hat sich eingebürgert auch im Deutschen de Begriff OU (bzw. im Plural OUs) zu verwenden

Ein weiterer wichtiger Punkt ist die Virtualisierung von Servern sowie Arbeitsplatzrechnern. Virtualisierung dient unter anderem - durch Konsolidierung von mehreren Hardwareplattformen auf eine einzige - der besseren Ressourcenauslastung. So ist es möglich, auch komplexe Testanlagen, die der produktiven Echtumgebung entsprechen, auf einer geringen Anzahl von physikalischen Maschinen zu realisieren. Dies soll hier einen Schwerpunkt des Einsatzgebietes für Virtualisierung bilden, die Abbildung einer möglichst realitätsgetreuen Testumgebung, um dort neue Softwareverfahren zu testen.

Ein weiterer Schwerpunkt ist die dynamische und schnelle Bereitstellung von virtuellen Terminalservern, die quasi auf Abruf in Betrieb genommen werden können. Diese Server "on-demand" werden vorzugsweise dazu genutzt, Software und Installation auf virtuellen Terminalservern in realer Produktivumgebung zu verifizieren. Dies ist in der Regel nach einer ersten Evaluierung in einer Testumgebung der nächste Test in der Echtumgebung. Dabei werden zusätzliche Ressourcen - vorzugsweise Terminalserver und Clients - bereitgestellt, mit denen ausgewählte Testbenutzer in der Echtumgebung neue Produkte und Verfahren auf ihre Tauglichkeit testen. Vergleichbares gilt für zusätzliche virtuelle Endgeräte am Arbeitsplatzrechner des Administrators.

Vorbemerkung

Dieser Arbeit ist ein Datenträger mit Quellen, Internetverweisen, Quellcode und Konfigurationsdateien beigefügt. Die meisten als Referenz angegebenen Links sind auch als PDF-Datei beigefügt, falls ein Link beim Lesen nicht mehr verfügbar sein sollte.

Im Rahmen dieser Arbeit verwendete IP-Adressen und Passwörter werden verfälscht, da es sich hierbei um vertrauliche Daten handelt und diese Diplomarbeit auch außerhalb der Kreisverwaltung Mayen-Koblenz zugänglich ist. Hier verwendete Netzwerkübersichten sind absichtlich einfach gehalten und zeigen nur den prinzipiellen Aufbau des Netzes. Aus Gründen der Sicherheit sind bewusst sämtliche sicherheitsrelevanten Maßnahmen (Firewall, Gateways, Antiviren- und Spamfilter, Intrusion Detection, Policy Based Rules, IEEE 802.1x, Netzübergänge, etc.) nicht aufgeführt.

Die wichtigsten im Rahmen dieser schriftlichen Ausarbeitung verwendeten Begriffe stellt ein Glossar in einer Übersicht zusammen. Im Anhang befindet sich weiterhin noch eine Tabelle, die die wichtigsten und häufigsten Dateiendungen kurz erläutert.

2 Problemstellung

Als 1999 der Siegeszug der PCs über die AS/400-Terminals in der Kreisverwaltung Mayen-Koblenz begann und der flächendeckende Einsatz der PCs weiter fortschritt, wurden diese PCs über Jahre verteilt beschafft. Die AS/400 ist ein Midrange Server System von IBM mit einem proprietären Betriebssystem von IBM.
Zeitgleich begann die Ablösung der AS/400-Terminals durch eine Vielzahl von Thin-Clients, die über Terminalserver eine Windows-Oberfläche mit Office-Paket anboten. Die Bedienung der weiterhin vorhandenen AS/400 wurde von den Terminalservern über Terminalemulationssoftware (IBM Client Access) sichergestellt und war somit für jeden Client verfügbar. Bis zum Jahre 2005 waren es knapp 400 Geräte insgesamt. Davon ca. 220 Thin-Clients und ca. 180 PCs. Im Bereich der Thin-Clients haben sich bis dahin 4 verschiedene Typen der Hersteller HP und Wyse etabliert. Im PC-Bereich waren es 17 verschiedene Rechnertypen der Hersteller IBM, Fujitsu und Fujitsu-Siemens. Mangelnde gute und zentrale Verwaltungsmöglichkeit der Thin-Clients sowie die Hardware-Typenvielfalt und Inhomogenität der Installationen der PCs machten den Verwaltungsaufwand für diese Geräte immens hoch. Zudem wies der Großteil der in die Jahre gekommenen Hardware eine hohe Ausfallrate auf und zog teure Reparaturen nach sich, die oftmals mit zeitaufwändigen Neuinstallationen verbunden waren.

Zwar waren Ansätze von Klonen vorhanden, also der Möglichkeit per Image-Software fertige und komplette Installationen von Betriebssystemen und Anwendungssoftware auf Hardware gleicher Typen zu "kopieren" anstatt manuell zu installieren. Aber diese prinzipielle Möglichkeit wurde erschwert durch die große Anzahl verschiedener Rechnertypen, für die je ein hardwareabhängiges Image vorgehalten werden musste. Zwar mussten nicht 17 verschiedene Images vorgehalten werden, da es durchaus möglich war, ein und dasselbe Image auf mehrere ähnliche Hardwarekonfigurationen aufzuspielen, aber ein manuelles Nachkonfigurieren diverser Komponenten - wie Grafik-, Sound- und Netzwerkkarte - war in vielen Fällen notwendig.

Weiterhin war keine zentrale Pflege der Images etabliert, die dafür sorgte, dass die unterschiedlichen Images dieselbe Software enthielten. Das Fehlen einer Standardisierung sowie die vorhandene Inhomogenität in Bezug auf Gleichheit der Installation machten natürlich auch das Troubleshooting, also die Analyse und die Problemeingrenzung zur Lösung von EDV-Problemen, schwierig und ineffektiv. Eine Gleichartigkeit der Umgebung war quasi nicht gegeben. Der Break-Even-Point zwischen Fehlersuche und einer Neuinstallation (nach erfolgloser Fehlersuche) war schlecht. Ein heute durchaus üblicher Wert von durchschnittlich 30-60 Minuten, um ein Problem zu beheben oder aber der Rechner neu zu installieren, war bei weitem nicht denkbar. Dies führte dazu, dass üblich war, von jedem einzelnen Rechner ein eigenes - sozusagen "persönliches" Image - zu erstellen, das im Fehlerfall eingespielt werden konnte.

Aber auch die ständige Aktualisierung dieser individuellen Images war sehr zeitaufwändig und ineffektiv. Weiterhin konnten die Images nur durch Booten des Rechners mit einer netzwerkfähigen Diskette zurückgespielt werden. Es gab zwar eingeschränkt die Möglichkeit Disketten zu erstellen, die mehrere Treiber enthielten, doch der Wildwuchs an Disketten war unvermeidlich. Zudem setzte die KVMYK in den PCs Netzwerkkarten auf Glasfaserbasis ein. Nicht für alle diese Netzwerkkarten gab es DOS-Treiber. So war es unter Umständen oft erforderlich, erst eine zusätzliche Netzwerkkarte einzubauen oder die Festplatte umzubauen, um ein Image zu erstellen oder zurück zu spielen. Auch ein Booten über PXE (Pre-Execution-Environment) war mit diesen Netzwerkkarten nicht möglich.
Dieses Problem wurde im Vorfeld dieser Diplomarbeit aus der Welt geschafft, indem die Anbindung von Büros an glasfaserfähigen Switches endet, die darüber hinaus einen integrierten Miniswitch mit mehreren 10/100 TX-Ports enthalten, an den die Endgeräte angeschlossen werden.

Ähnliche Probleme gab es mit den Terminalservern. Diese sind bezüglich Softwareinstallationen mit den normalen PCs vergleichbar. Schließlich wird die Anwendungssoftware wie bei einem PC installiert und den Benutzern an Thin-Client-Geräten über eine Terminalserversitzung zugänglich gemacht, als ob sie an einem PC arbeiten würden.

Sämtliche Maschinen (PCs und Server), die Microsoft Windows NT Betriebssysteme einsetzen, waren in der KVMYK in einer Windows NT Domäne organisiert. Eine NT Domäne fasst Objekte wie Server, Domänencontroller, Computer, Benutzergruppen sowie Benutzerkonten zu einer zentral administrierbaren Einheit zusammen. Diese Objekte werden in einer Datenbank, der SAM-Datenbank (Security Account Manager), verwaltet, die auf den Domänencontrollern der Domäne gehostet wird. Innerhalb dieser SAM-Datenbank werden die Objekte nicht mit ihren Namen, sondern mit einer eindeutigen Kennung, der sogenannten SID (Security Identifier), identifiziert. Diese eindeutige SID wird bei der Installation des Betriebssystems erzeugt. Wird der Computer nun Teil einer Domäne, werden ihn eine Kombination aus SID und Domänenidentifier eindeutig in der Domäne kennzeichnen. Ein solcher Domänencomputer kann geklont und umbenannt werden, aber die SID bleibt davon unverändert. Werden also keine geeigneten Maßnahmen getroffen, tauchen so zwei oder mehr Computer auf, die innerhalb der Domäne nicht mehr eindeutig zu identifizieren und damit voneinander unterscheidbar sind. Dies kann zu Inkonsistenzen und gravierenden Problemen führen. Ein Drittanbietertool kann hier Abhilfe schaffen, indem eindeutige voneinander unterschiedliche SIDs für jeden Computer erzeugt werden.

Ähnlich organisiert ist auch die sogenannte Terminalserverfarm (entspricht in etwa der Windows-Domäne), in der die Terminalserver mit Citrix Metaframe bzw. Citrix Presentation Server verwaltet werden. Auch diese besitzen einen eindeutigen Identifier, der beim Klonen ebenfalls ohne geeignete Eingriffe zu Inkonsistenzen innerhalb der Farm führen kann. Leider gibt es hier kein Drittanbietertool, so dass an dieser Stelle manuell gearbeitet werden muss. Diese vorgenannten Probleme sorgten dafür, dass auch von jedem einzelnen Terminalserver ein eigenes Image gepflegt werden musste. Auch hier entstanden im Laufe der Zeit immer wieder Diskrepanzen in den Softwareinstallationen und stellten schwierige Verhältnisse bei der Fehlersuche dar. Erschwerend kam hinzu, dass die Benutzer innerhalb der Farm mit einem Lastausgleich (Load Balancing) automatisch und gleichmäßig auf die verschiedenen Terminalserver verteilt wurden. So sind Fehler, die nur an einem dedizierten Server auftreten, schwieriger zu diagnostizieren. Also ist hier die Homogenität besonders wichtig.

Dies alles sorgte für einen hohen Grad der Auslastung der IT Mitarbeiter und für die übermäßige Beanspruchung von Ressourcen für die Verwaltung und Einhaltung der Konsistenz der IT Infrastruktur.

War ein Hardwarefehler eines Thin-Clients durch einen einfachen Austausch der Hardware relativ schnell behoben, so war dies bei einem PC-Arbeitsplatz nicht so. Bei einem Thin-Client dient die Hardware und ein kleines Stück Standardsoftware lediglich der Anbindung an den Terminalserver. Daher ist dieser komplette Arbeitsplatz verhältnismäßig schnell austauschbar. Im Gegensatz dazu steht ein PC-Arbeitsplatz mit seiner umfangreichen lokal installierten Software. Hier dauerte die Wiederherstellung der Arbeitsfähigkeit wesentlich länger. Hier ist es wichtig zunächst dafür zu sorgen, dass möglichst viel auf Terminalservern und möglichst wenig auf lokale PCs installiert wird. In vielen Fällen erweist sich dann ein PC-Arbeitsplatz als obsolet, so dass dauerhaft auf einen Thin-Client gewechselt werden kann. Zum anderen kann eine Vereinheitlichung und Angleichung von Thin-Client- und PC-Benutzern dafür sorgen, dass ein Ausfall eines PCs durch das Aufstellen eines Thin-Clients kompensiert werden kann, so dass die Arbeitsfähigkeit des Arbeitsplatzes zu großen Teilen innerhalb kürzester Zeit wiederhergestellt werden kann. Hier stehen Standardisierung, Homogenisierung und Flexibilität im Mittelpunkt.

In drei von vier Außenstellen war jeweils eine eigene lokale Serverinfrastruktur, bestehend aus File- und ggf. Mailserver, zu betreuen. Weiterhin waren nach dem bestehenden Konzept in Außenstellen immer zwingend PC-Arbeitsplätze vorgesehen. Allein hierdurch waren ca. 75 Fat-Clients notwendig.

Ein Ausweg aus dieser gewachsenen Infrastruktur und deren typischen Problemen war eigentlich nur denkbar, indem eine stärkere Standardisierung, Zentralisierung und ein hoher Grad an Automatisierung eingeführt werden. Diese Gelegenheit bot sich im Rahmen der bevorstehenden Migration des Betriebssystems Windows NT auf Windows Server 2003 bzw. Windows XP und der Migration von Citrix Metaframe 1.8 auf Citrix Presentation Server 4.0.

3 Aufgabenstellung und Zielsetzung

IT-Migration heißt, dass heterogene und/oder technisch überalterte Systeme neuen IT-Lösungen zugeführt werden. IT-Migration bedeutet zugleich einen Veränderungsprozess in den bestehenden Geschäftsabläufen sowie der eingesetzten Technologie. Die Aufgabenstellung und die zu lösenden Probleme waren Teil des Migrationsprojektes der KVMYK. Der Schwerpunkt dieser Arbeit liegt hier im Bereich Standardisierung, Homogenisierung, Automatisierung und Flexibilisierung mit dem Ziel der (voll-)automatisierten Installation von Endgeräten - sowohl Thin- als auch Fat-Clients - sowie von Servern. Ein weiterer Punkt liegt in der Übertragung der gewonnen Erkenntnisse in die Virtualisierung von physikalischen Maschinen zu Testzwecken. Kann oder will ein Softwarehersteller die Terminalservertauglichkeit seines Produktes nicht garantieren, so ist im Vorfeld eine Evaluierung unvermeidbar, um sicherzustellen, dass die Software unter Terminalservern betrieben werden kann. Bei umfangreichen Lösungen bietet sich dafür eine Testumgebung an, bei kleineren Testzeremonien ist ein Testserver in der Echtumgebung ausreichend. Im Weiteren dient die Testanlage auch zur Evaluierung von Veränderungen innerhalb des Active Directory.

3.1 Ausgangslage

Die KVMYK hat ihren Hauptsitz in der Bahnhofstraße 9 in Koblenz. Die vier Außenstellen befinden sich in Koblenz, Andernach und zweimal in Mayen. Nach anfänglicher Analyse zum Start der Migration am 01. Oktober 2005 stellte sich die in den nachfolgenden Kapiteln genannte Situation dar. Fast jeder der insgesamt ca. 400 Mitarbeiter besaß einen eigenen EDV-Arbeitsplatz und die Verteilung der Mitarbeiter stellte sich wie folgt dar:

Standort	Mitarbeiter	Funktion
Koblenz, Bahnhofstraße 9	314	Kreishaus
Mayen, Siegfriedstraße 20	11	Kfz-Zulassung
Koblenz, Neversstraße 4	33	Gesundheitsamt Koblenz
Mayen, Bannerberg 9	15	Gesundheitsamt Mayen
Andernach, Breite Straße 109	5	Gesundheitsamt Andernach
Summe =	378	

Tabelle 3.1 Standorte und Mitarbeiter

Bis dahin wurde überwiegend Windows NT Workstation sowie Windows NT Server als Betriebssystem eingesetzt. Das Officepaket bestand aus Windows Office 97. Der Standard-

Mailclient war Outlook 2000. Die Anbindung an die vorhandenen zwei AS/400-Host-Systeme erfolgte über IBM Client Access. Der Browser war der betriebssystemeigene Internet Explorer 5. Als Anzeigeprogramm für Dateien im PDF-Format (Portable Document Format) wurde der Reader von Adobe in der Version 5 eingesetzt.

3.1.1 Allgemeines/Übersicht

Die folgenden Tabellen und Übersichten sollen einen ersten Eindruck von dem Umfang der Migration vermitteln.

Standort	Endgeräte	Server	Anmerkung
Koblenz, Bahnhofstraße 9	368	25	
Mayen, Siegfriedstraße 20	12	0	Keine lokale Datenhaltung vor Ort
Koblenz, Neversstraße 4	35	2	
Mayen, Bannerberg 9	17	1	
Andernach, Breite Straße 109	5	1	
Summe	437	30	

Tabelle 3.2 Anzahl der Endgeräte/Server pro Standort

Die deutlich höheren Gerätezahlen im Vergleich zu den Mitarbeiterzahlen ergeben sich dadurch, dass z.B. in den Gesundheitsämtern ärztliche Untersuchungszimmer zusätzlich mit Geräten ausgestattet sind. Im Weiteren sind noch Leihlaptops, allgemein zugängliche Geräte, z. B. für die Bücherei oder Geräte für den Katastrophen- bzw. Seuchenfall sowie Schulungsraum und Arbeitsplätze für Nachwuchskräfte und Praktikanten zu nennen. Im Gegensatz dazu gibt es nur wenige Arbeitsplätze, an denen sich mehrere Mitarbeiter ein Gerät teilen (Cafeteriapersonal, Fahrer und Information).
In der Kraftfahrzeugzulassung am Standort Mayen erfolgt keine lokale Datenhaltung, sondern die Arbeitsplätze sind direkt an das Kreishaus angebunden. Die folgende Tabelle verdeutlicht die Anbindung der Außenstellen an die Zentrale im Kreishaus Koblenz.

von Standort	zu Standort	Anbindung
Mayen, Siegfriedstraße 20	Koblenz, Bahnhofstraße 9	2 MBit/s
Mayen, Bannerberg 9	Mayen, Siegfriedstraße 20	2 MBit/s
Koblenz, Neversstraße 5	Koblenz, Bahnhofstraße 9	2 MBit/s
Andernach, Breite Straße 109	Koblenz, Bahnhofstraße 9	128 KBit/s

Tabelle 3.3 Anbindung der Außenstellen an die Zentrale

Die Verteilung der Geräte nach Thin- bzw. Fat-Client und dem darauf enthaltenen Betriebssystem geht aus nachstehender Übersicht hervor.

Endgerätetyp	Anzahl	Betriebssystem
Thin-Client	220	Windows CE 3 bzw. 4
Fat-Client	137	Windows NT 4.0 Workstation
Fat-Client	35	Windows XP Professional
Fat-Client	5	Windows 2000 Professional

Tabelle 3.4 Verteilung der Betriebssysteme bei Thin- bzw. Fat-Client

Die Endgeräte mit Windows XP Professional waren bereits im Einsatz, weil entweder Fachverfahren oder notwendige Hardware von Seiten Windows NT nicht mehr unterstützt wurden. Ansonsten wurde aber auch auf den Windows XP Professional Rechnern die gleiche Standardsoftware eingesetzt wie auch auf den NT-Rechnern.
Die folgende Übersicht zeigt das Netz der Kreisverwaltung Mayen-Koblenz. Aus Gründen der Übersichtlichkeit wurden in den Außenstellen die Endgeräte und ggf. vorhandenen Server nicht dargestellt.

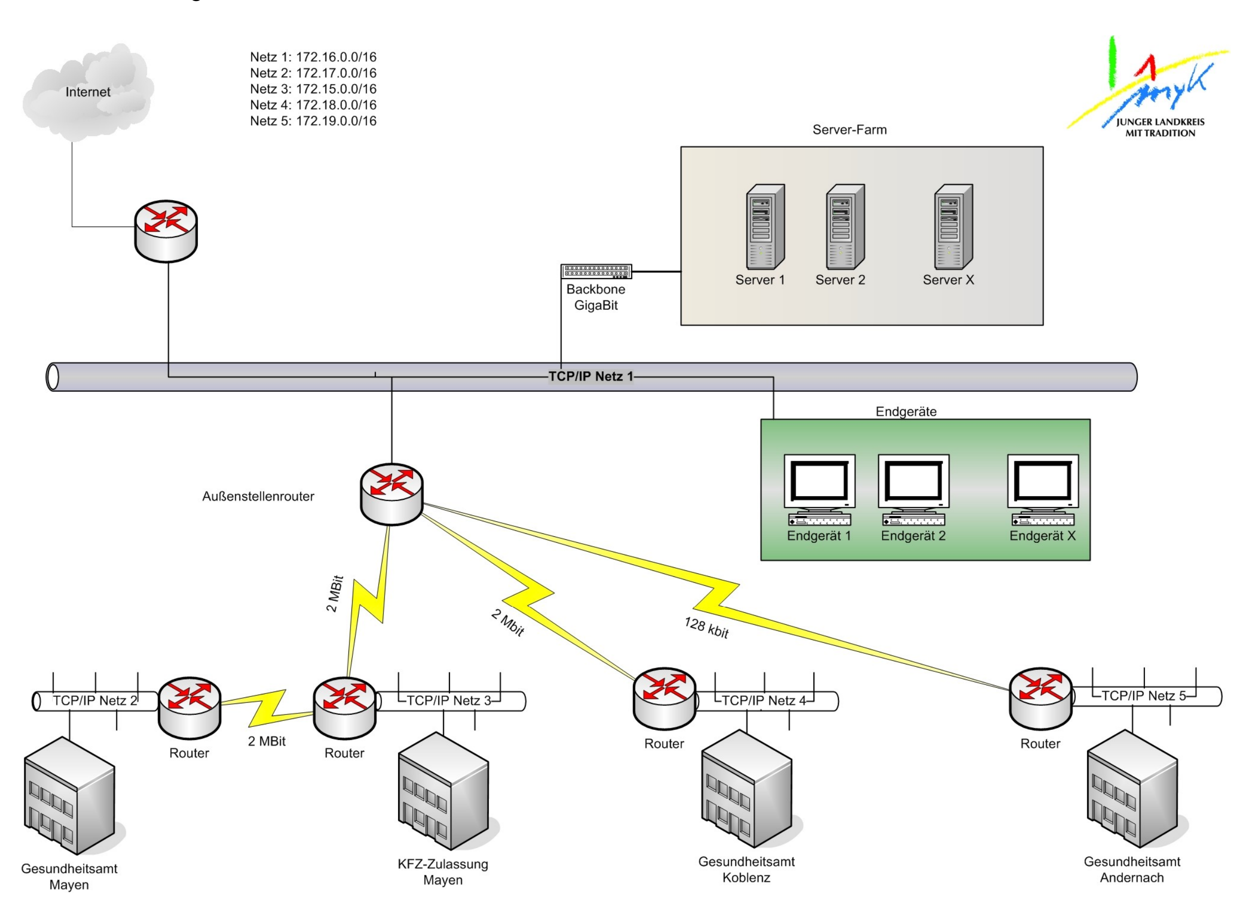


Abbildung 3.1 Netzwerkdiagramm der KV Mayen-Koblenz

3.1.2 Gründe einer Migration

Ein Wechsel des Betriebssystems und der Officekomponenten wurde unter anderem aus folgenden Gründen notwendig:

- Microsoft stellte bereits Mitte 2004 den Support/Unterstützung für das Betriebssystem Windows NT Workstation ein. Dieses Betriebssystem wurde auf fast allen Arbeitsplatz PCs eingesetzt.
- Microsoft stellte zum 31.12.2004 den Support/Unterstützung für das Betriebssystem Windows NT Server und Windows NT Server Terminalserveredition[3] ein. Diese Betriebssysteme wurden nahezu auf allen Servern sowie Terminalservern eingesetzt.
- Neue Hard- und Software werden zukünftig nicht oder nur ohne Gewähr die eingesetzten Betriebssysteme Windows NT unterstützen.
- Die eingesetzte Hardware ist größtenteils veraltet (Betriebsdauer mehr als 5 Jahre) und den heutigen Anforderungen nicht gewachsen.
- Microsoft stellte Anfang 2004 den Support/Unterstützung für Office 97 ein.

Neben der oben genannten fehlenden Weiterentwicklung und Unterstützung durch die Firma Microsoft sprachen auch andere zwingende Gründe für eine Migration. Die fortschreitende Verbreitung von USB-Geräten (z.B. Drucker, Scanner und Kameras) machten es nahezu unmöglich, weiterhin ohne USB-Unterstützung zu arbeiten. Viele Softwarehersteller machten sich zudem nicht mehr die Mühe ihre Produkte auf Kompatibilität zu testen, geschweige denn Support für ihre Software unter NT zu leisten.

Im Rahmen einer Entscheidungsfindung für ein Windows oder Open Source Betriebssystem fiel die Wahl aufgrund eines sehr hohen Grades an Integration von Microsoft-Lösungen auf eine vollständige fortführende Migration. Damit stand das Zielbetriebssystem Windows Server 2003 bzw. Windows Professional XP mit dem Officepaket 2003 fest. [Tho05]

Die Anforderungen, die ein neues Betriebssystem stellte, machten außerdem den Austausch des Großteils der Hardware erforderlich. Im Bereich der Server mussten alle wichtigen Server (Domänencontroller, File-, Print-, Mail- und Terminalserver) erneuert werden, da die Hardware zwischen 5 und 7 Jahre alt war. Für einzelne Server war auch seitens der Herstellerfirma keine Supportunterstützung mehr gewährleistet. Im Bereich der Clientgeräte wurde eine zweistufige Ablösung der Hardware notwendig. Sämtliche vorhandenen Thin-Clients waren zwischen 6 und 7 Jahre alt, ein Großteil der PCs ebenfalls. Diese wurden in einer Ausschreibung im Rahmen der Migration neu beschafft. Ein kleiner Teil von ca. 30 Geräten erfüllte die Minimalanforderungen von Windows XP Professional und wurde mit der Aussicht, diese Geräte noch ca. 12-18 Monate zu betreiben, in die Migrationsplanung mit aufgenommen.

3.1.3 Ziele der Migration

Grundsätzlich könnten im Rahmen einer IT-Migration lediglich die technischen Komponenten verändert werden (Austausch veralteter Hardware, Anhebung des Betriebssystems). Dies würde jedoch bedeuten, dass das aktuelle technisch-organisatorische Konzept ohne kritische Betrachtung und Neuausrichtung an den Zielen der Zukunft 1:1 portiert wird. Dies ist im Hinblick auf einen zukunftsorientierten und wirtschaftlichen Einsatz der IT nicht sinnvoll. Insoweit sollte mit der IT-Migration auch das organisatorische Umfeld überprüft und soweit notwendig verändert werden.

[3] Bei NT waren die Terminalservices noch ein eigenes Betriebssystem. Seit Windows 2000 sind sie jedoch als Dienst realisiert und jederzeit nachträglich installierbar.

Im Rahmen der Migration wurden die folgenden Ziele als die wichtigsten Punkte festgehalten, die zu erfüllen sind:

1. Wirtschaftlichkeit des IT-Einsatzes (Reduzierung der TCO)
2. Zukunftssicherheit im Hinblick auf absehbare Anforderungen
3. Konsolidierung und Homogenisierung der IT-Systemlandschaft
4. Vereinfachung der Systemadministration, -pflege und Benutzerbetreuung
5. Reduzierung von Verwaltungsaufwand
6. Hohe Benutzerfreundlichkeit und -akzeptanz der eingesetzten Technik/Software
7. Hohe Daten- und Ausfallsicherheit der Systeme

Ein besonderer Wert wird auf Flexibilität im Einsatz der Endgeräte gelegt. Einerseits wird ein und dieselbe Hardware zu unterschiedlichen Zwecken eingesetzt, als Thin- oder Fat-Client. Andererseits soll eine modulare Erweiterbarkeit der Endgeräte gewährleistet sein, z.B. nachträglicher Einbau eines DVD-R/W-Laufwerkes oder eines Scanners. Das Ziel, die Anzahl von 55% Thin-Clients auf 75% zu steigern, bedeutet bei einer Zahl von ca. 400 Arbeitsplätzen rund 80 Benutzer, die von einem PC auf einen Thin-Client umsteigen. Zum Zeitpunkt der Migration war der Großteil der vorhandenen Thin-Clients 7 Jahre alt. Diese echten Thin-Clients waren Geräte von HP und Wyse und hatten enorme Probleme in der grafischen Darstellung und größtenteils spürbare Verzögerungen bei Mausbewegungen und Tastendruck und somit allgemein ein schlechtes Antwort-Zeit-Verhalten. Um Akzeptanzprobleme bei diesen Thin-Client Benutzern zu verhindern, wurde die Hardware im Bereich der Endgeräte großzügig dimensioniert. [FhG05] Der geringe Preisunterschied[4] zwischen einem Thin-Client und einem Einsteiger-PC ermöglichte es als Endgeräte ausschließlich vollwertige PCs zu verwenden und auf den Einsatz von echten Thin-Clients zu verzichten.

Im Rahmen dieser Diplomarbeit liegt das Augenmerk bei den oben genannten Punkten 3., 4., 5. und 6. Die anderen Punkte sind prinzipiell davon genauso betroffen, da sie aus dem Gesamtprojekt nicht nur einzeln zu betrachten sind. In den nachfolgenden Kapiteln werden die Einzelziele noch den vorgenannten großen Migrationszielen zugeordnet.

3.2 Ziele der Arbeit

In den nachfolgenden Kapiteln werden die großen Ziele dieser Arbeit und damit einige Teilziele der Migration als Ganzes dargestellt. Sowohl Homogenisierung der Abbilder als auch die Automatisierung der Installationen senken die TCO, vereinfachen die Systemadministration und sorgen für eine Senkung des Verwaltungsaufwands. Die angestrebte Lösung soll aber auch zukunftssicher sein, vor allem im Bereich der Fat-Clients vor dem Hintergrund der neuen Betriebssysteme Windows Vista und Windows Server 2008, die erschienen sind bzw. in einer Betaversion vorliegen[5].

Im Hinblick auf Wirtschaftlichkeit und leere Kassen der öffentlichen Haushalte liegt ein Ziel in der Verwendung vorhandener Ressourcen für den Aufbau und den Betrieb der Testumgebung. Weiterhin ist eine elementare Forderung die flexible Verwendungsmöglichkeit der Endgeräte, d.h. der Einsatz als Thin- oder Fat-Client muss ohne Zusatzkosten möglich sein. Im Weiteren ist der Anspruch für die Investitionssicherung für 5 Jahre gegeben.

---

[4] Das Gerät aus dem damaligen Landesrahmenvertrag war sogar günstiger als ein vergleichbarer Thin-Client!

[5] Stand der Angaben ist Dezember 2007

3.2.1 Ziel: Homogenisierung der Abbilder

Bisher gab es aufgrund der inhomogenen und sehr unterschiedlichen Hardware sehr viele verschiedene Images. Zum einen war dies die Folge der bis zu 17 verschiedenen Rechnertypen, die eine große Vielfalt von Abbildern erforderte, zum anderen gab es keine Versionierung der Images. Die dezentrale Erstellung und Pflege der Images in Kombination mit mangelnder Dokumentation führte somit zu einer kaum beherrschbaren Flut an Images.

Homogenisierung meint an dieser Stelle die zentrale Erstellung, Pflege und Dokumentation der Images mit dem Ziel, möglichst mit einem einzigen Image alle vorhandenen Rechnertypen und Softwareforderungen abzudecken. Die Homogenisierung soll sowohl im Bereich der Endgeräte als auch im Bereich der Terminalserver erreicht werden.

3.2.2 Ziel: Automatisieren der Endgeräteinstallationen

In der vorhandenen alten Umgebung beinhaltete das Ausbringen eines Images die folgenden Schritte, die von einem Administrator ausgeführt werden mussten.

1. Booten des Zielrechners mit einer netzwerkfähigen Diskette.
2. Manuelles Einloggen mit Benutzername und Kennwort in der Domäne und Verbinden der benötigten Netzwerklaufwerke.
3. Aufrufen des Imageprogramms mit den erforderlichen Parametern.
4. Nochmaliges händisches Booten des Zielrechners mit einer weiteren Diskette, die ein Programm beinhaltet, um den Rechnernamen manuell anzupassen und die SID automatisch zu verändern.
5. Weiteres Booten des Zielrechners und manuelle Anmeldung als lokaler Administrator mit dem Ziel, den Rechner in die Domäne zu bringen. Anschließend ist ein Neustart erforderlich.
6. Ggf. weitere manuelle Anpassungen, die sich aus der Namensänderung des Rechners ergeben, wie z.B. Änderungen am Citrix Terminalserver Client.

Die Automatisierung hat hier zum Ziel, dass einerseits die Installation, also das Ausbringen des Images auf den Zielrechner, nicht von einem Administrator ausgeführt werden muss, sondern im Idealfall auch von dem Benutzer selbst durchgeführt werden kann. Zum anderen sollen die manuellen Eingaben möglichst entfallen, da sie einerseits in der Regel administrative Berechtigungen erfordern und andererseits fehleranfällig sind. Sowohl bei der Erstellung als auch bei der Wiederherstellung eines Rechners soll der Benutzer am Ende des Prozesses in der Lage sein, sich wieder an seinem Endgerät anzumelden und damit zu arbeiten.

Daraus ergibt sich eine neue Ablaufreihenfolge, die von einem Benutzer durchgeführt werden kann.

1. Der Benutzer tritt mit der Administration in Verbindung, wenn mit dem Rechner des Benutzers ein Image erstellt oder wiederhergestellt werden soll.
2. Nach Aufforderung durch einen Administrator bootet der Zielrechner mit einem netzwerkfähigen integrierten Medium, das durch den Benutzer mit einem speziellen Tastendruck ausgelöst wird.
3. Das Image wird automatisch erstellt oder wiederhergestellt und der Benutzer kann sich nach Beendigung des Vorgangs ohne weitere manuelle Eingriffe am Rechner anmelden.

Ein weiterer Vorteil, der sich automatisch ergibt, ist die wesentlich erhöhte Geschwindigkeit, da Zugriffe auf Diskettenlaufwerke im Vergleich zu Netzwerken erheblich langsamer sind. Generell soll aber der Vorgang als Ganzes beschleunigt werden und das nicht nur ausschließlich durch den Wechsel des Mediums.

3.2.3 Ziel: Automatisieren der Terminalserverinstallationen

Bei Serverinstallationen, insbesondere Terminalserverinstallationen mit der Terminalserver-Software Citrix, ergeben sich ähnliche Anforderungen, wie sie schon beim Endgerät vorliegen. Natürlich wird an dieser Stelle die Installation der Images und der Server nur durch die Administratoren erfolgen, Benutzer sind an diesem Vorgang nicht beteiligt. Eine weitere Problematik liegt hier in der Software Citrix. Diese benutzt, ähnlich den SIDs des Microsoft Betriebssystems, eigene eindeutige Identifier innerhalb der Farm von Terminalservern. Deshalb kann eine vorhandene Programminstallation nicht Teil eines Images sein, sondern muss nachträglich für jeden Zielserver erfolgen.
Hieraus ergibt sich eine Ablaufreihenfolge, die von einem Administrator durchgeführt werden soll.

1. Bei Bedarf wird ein Terminalserverimage erstellt oder wiederhergestellt.
2. Der Administrator bootet den Zielserver mit einem netzwerkfähigen integrierten Medium, das mit einem speziellen Tastendruck ausgelöst wird.
3. Das Image wird automatisch erstellt oder wiederhergestellt, im Anschluss wird automatisch die Terminalserverinstallation durchgeführt und Benutzer können sich nach Beendigung des Vorgangs ohne weitere manuelle Eingriffe am Terminalserver anmelden.

3.2.4 Ziel: Automatisieren der Installationen in einer virtuellen Umgebung

Der Begriff Virtuelle Umgebung ist hier so zu sehen, dass es sich vorwiegend um eine virtualisierte Testanlage handelt, in der möglichst automatisiert installiert werden kann. Auch hier handelt es sich bei der Automatisierung nur um Endgeräte - also Thin- und Fat-Clients - sowie Terminalserver. Portiert werden die in den vorgehenden Kapiteln erarbeiten Konzepte und Methoden auf eine virtuelle Infrastruktur. Dabei sollen Gemeinsamkeiten und Unterschiede verdeutlicht werden. Die Testanlage soll aber komplett ausgestattet sein und der Echtumgebung in ihrem Verhalten möglichst ähnlich sein.
In der produktiven Umgebung liegt das Augenmerk auf dynamischer und schneller Bereitstellung von virtuellen Terminalservern. Dafür wird ein separater dedizierter leistungsfähiger Host vorgehalten, der die produktiven virtuellen Maschinen beherbergt. Die virtuellen produktiven Terminalserver dienen in erster Linie zur Verifikation von neuen Verfahren und dazu notwendigen Softwareinstallationen sowie zur Erprobung von Patches und Updates. Ist dieser Test bestanden, werden die Installationen auf die produktiven realen Terminalserver übertragen.
Eine weitere Anwendung virtueller Clients ist bei den Administratoren zu sehen. Diese besitzen einen administrativen PC, der von den Standardgeräten abweicht. Neue Software auf PCs zu testen oder im Fehlerfalle ein vorhandenes Szenario des Endanwenders nachzustellen, macht das Vorhandensein eines zusätzlichen realen Rechners überflüssig. Dieser kann als virtueller Client im Rahmen der kostenlosen und frei verfügbaren Software "VMware Server" am Arbeitsplatz des Administrators betrieben werden.

3.3 Aufbau der Arbeit

Die Arbeit beschäftigt sich im nächsten Kapitel mit dem Stand der Technik in organisatorischer und technischer Hinsicht und fasst die Probleme zusammen, die zur Erreichung der geforderten Ziele gelöst werden müssen.
Im Kapitel "Lösungsansatz" werden Lösungen der organisatorischen und technischen Probleme vorgeschlagen, die in den darauffolgenden Kapiteln detailliert dargestellt und erläutert werden. Dem kleineren aber nicht weniger wichtigen Kapitel zu den organisatorischen Arbeiten folgt die technische Aufarbeitung des Lösungsvorschlags. Dabei werden standardisierte Dienste, wie DHCP und PXE, und auch selbst entworfene Scripts als Mittel zum Zweck genau erläutert.
Im Kapitel 8 erfolgt die Fortführung der erarbeiteten Lösungen in die virtuelle Umgebung. Dabei muss die vorhandene Hardware möglichst optimal genutzt werden.
Kapitel 9 stellt das Ergebnis aus geforderten und erreichten Zielen dar und das daran anschließende Kapitel 10 fasst die Arbeit kurz zusammen.
Ein Ausblick auf wünschenswerte Erweiterungen oder Eigenschaften schließt die Arbeit ab.
Im Anhang bzw. auf einem beigefügten Datenträger befinden sich sämtliche Software-Quellen und Links, bei kostenfreien Produkten auch die Installationsdateien. Des Weiteren sämtliche verwendeten und angesprochenen Scripte sowie die verwendeten Konfigurationsdateien und das erstellte Perl-Script samt Interpreter. Einige Printscreens schließen den Anhang ab.

4 Stand der Technik

Im zeitlichen Bezug befinden wir uns hier etwa Ende 2005. Zu diesem Zeitpunkt sind die Weichen für die spätere Migration gestellt und aufgrund der seinerzeit vorliegenden technischen und organisatorischen Rahmenbedingungen Entscheidungen getroffen worden. In einer Sitzung des Kreisausschusses wurde im Juni 2005 beschlossen, dass

1. eine fortführende Migration zu Windows Server 2003 bzw. Windows XP,
2. eine Fortführung des Einsatzes von OSS-Produkten in Nischenbereichen und
3. eine Untersuchung der Ablösung von Thin-Clients auf Basis von OSS-betriebener Geräte

durchgeführt werden soll.

Betriebssystem

Windows Server 2003 bzw. Windows XP waren zum offiziellen Start des Migrationsprojektes im Herbst 2005 ca. 2 Jahre im Einsatz. Dementsprechend stabil war das Betriebssystem nach Veröffentlichung von Service Pack 2 für Windows XP im August 2004. Ein vergleichbar stabiles Serverprodukt wurde Ende 2005 mit der Erscheinung von Windows Server 2003 R2 veröffentlicht. Bezüglich technischer Fehlerbehebung entsprach Windows Server 2003 R2 der Version mit installiertem Service Pack 1. Allerdings wurde von Seiten von Microsoft diese Version R2 als völlig neue Lizenz verkauft, d.h. mit einer vorhandenen Lizenz für Server 2003 war es nicht erlaubt Server 2003 R2 zu installieren. Dies wurde u. a. mit technischen Erweiterungen von R2 begründet, wie umfassende Verbesserungen in den Bereichen Identitäts- und Zugriffsverwaltung, Serververwaltung in Zweigstellen, Speichereinrichtung und -verwaltung sowie Anwendungsentwicklung.

Anwendungssoftware

Die Wahl für ein neues Officeprodukt fiel auf Windows Office 2003. Auch hier waren ähnliche Faktoren entscheidend wie für die Findung der Version des Betriebssystems, also eine Markteinführung vor mindestens 2 Jahren. Damit konnte auf ein ausgereiftes Produkt zugegriffen werden, das in der Version mit Service Pack 2 relativ fehlerfrei und für den produktiven Einsatz auf Terminalservern als geeignet angesehen werden konnte. Wichtig bei dieser Entscheidung war ebenfalls die Möglichkeit den Unterschied zum bisher eingesetzten Microsoft Office 97 so gering wie möglich zu halten. Damit sollten der Schulungsaufwand minimiert und die Umgewöhnung der Endanwender erleichtert werden, um so eine höhere Akzeptanz und Zufriedenheit der Anwender zu erreichen. Ein weiteres entscheidendes Kriterium war der hohe Grad der Verflechtung zwischen Fachsoftware und dem Microsoft Officeprodukt. Inzwischen existieren in Office über 2000 selbst erstellte Vorlagen. Viele Hersteller benutzen Officeprodukte, sei es zur optischen Aufbereitung der Daten, zum Export oder zur Druckausgabe. Wegen der großen Verbreitung von Microsoft setzen viele daher auf das

Produkt dieses Herstellers. Wegen der noch viel geringeren Verbreitung von Alternativen zu Microsoft Office machen sich viele Softwareanbieter nicht die Mühe, andere Officevarianten zu unterstützen. Zu dem gleichen Ergebnis kommt auch eine Studie des Fraunhofer Instituts für Arbeitswirtschaft und Organisation (IAO) aus dem Jahre 2006. [FIAO06]

Gerade im Hinblick auf den Einsatz von Software auf Terminalservern ist ein eher konservativer Ansatz der Auswahl von Softwareprodukten sinnvoll. Dies wird einsichtig, wenn man sich vor Augen führt, dass bis zu 50 Benutzer gleichzeitig auf einem Server arbeiten, und dass durch einen Softwarefehler verursachten Absturz des Servers direkt eine große Anzahl an Benutzern betroffen ist. Neben den vorgenannten Produkten führte dies zur folgenden Auswahl, die als Standardprodukte für die Arbeitsstationen und Terminalserver definiert wurden:

- Microsoft Office 2003 Professional (Officepaket)
- Outlook 2003 aus dem Paket Microsoft Office 2003 (Mail-Client)
- Internet Explorer 6 (Browser)
- Sun Java Runtime Environment 4 (Browsererweiterung und Java-Plattform)
- Adobe Acrobat Reader 7 (PDF-Betrachter)
- FreePDFXP 3 (PDF-Drucker)
- Irfanview 3 (Bildbetrachter)
- Pack- und Entpackfunktionalität über die betriebssystemeigene Komponente (zip)
- Citrix ICA-Client 9 (Zugriff auf Terminalserver bzw. Pass-Through auf weitere Terminalserver)
- Snapshot (zur komfortablen Erstellung von Printscreens)

Terminalserverclient

ICA ist ein Protokoll für Terminalserver, das von der Firma Citrix entwickelt worden ist. Der ICA-Client wird auch auf den Terminalservern selbst gebraucht, wenn z.B. ein Benutzer auf einem Terminalserver eine Anwendung auf einem anderen Terminalserver starten will. Dazu wird die Passthrough-Authentifizierung eingesetzt, d.h. der Benutzer wird aufgrund seiner aktuellen Anmeldeinformationen (Benutzername und Kennwort) automatisch authentifiziert. Diese Authentifizierungsmethode wird in der vorliegenden Konfiguration in der KVMYK oft eingesetzt. Ein Thin-Client startet immer primär einen Windows-Desktop, der von einem bestimmten Anteil der Terminalserver der Citrix-Farm, den sogenannten Desktop-Terminalservern, zur Verfügung gestellt wird. Diese Server haben ausschließlich die Aufgabe, einen Desktop samt den oben genannten Standardanwendungen zur Verfügung zu stellen. Sogenannte Sonder- oder Fachanwendungen, also alles, was nicht zum Umfang der Standardanwendungen gehört, werden auf zusätzlichen Terminalservern abgearbeitet. Diese Sonderterminalserver bieten keinen Windows-Desktop zum Login an, sondern der Benutzer startet eine Sonderanwendung so, als ob sie auf seinem lokalen Desktop installiert wäre. Wird eine solche Anwendung von einem Benutzer gestartet, wird eine weitere Sitzung auf einem zusätzlichen Terminalserver unter Ausnutzung der Passthrough-Authentifizierung initiiert und die Sonderanwendung gestartet. Der ganze Vorgang kann so eingestellt werden ("seamless window"[6]), dass der Benutzer (optisch) gar nicht merkt, dass er eine Anwendung auf einem weiteren Server gestartet hat. Der ganze Vorgang läuft für den Anwender vollkommen transparent ab, d.h. er startet die Anwendung wie jede andere und kann sämtliche Aktionen durchführen, die mit lokalen Anwendungen auch möglich sind, also z.B. das Mi-

[6] Bedeutet die nahtlose Integration in die vorhandene Umgebung.

nimieren in der Taskleiste[7]. Gleiches gilt auch für die Fat-Clients, d.h. starten diese eine Anwendung über den Terminalserverclient, verhält sich diese Anwendung so, als ob sie auf dem lokalen PC gestartet worden wäre.

Open Source Software

Die geforderte Untersuchung der Ablösung der vorhandenen Windows CE Thin-Clients hin zu OSS basierenden Geräten brachte die Entscheidung zu Gunsten des Betriebssystems eLux NG der Firma Unicon. Diese COLS-Distribution in Verbindung mit der Verwaltungssoftware Scout NG brachte die besten gesamtwirtschaftlichen Ergebnisse, die u. a. begründet sind in

- einer zentralen und einfachen Verwaltungskonsole (an MS AD angelehnt),
- MS Active Directory-Unterstützung,
- der Unterstützung von Fat-Clients,
- einer schnellen und kleinen Distribution (Eignung für WAN-Strecken),
- der Unterstützung der Installation über PXE,
- Datenbankunterstützung, SQL-Server, MySQL,
- Multiple Administrator Policy[8],
- Script Command Schnittstelle und offene DB-Schnittstelle,
- Hochverfügbarkeit und Skalierbarkeit der Serverkomponenten,
- Inkrementelles Firmwareupdate.

Die Scout Software besteht aus den folgenden vier Komponenten:

- Scout Enterprise Server
- Scout Enterprise Konsole
- Container
- ELIAS NG

Die zentrale Komponente des Scout Servers ist eine Datenbank, die über die Informationen der aktuell verwalteten Geräte verfügt, über deren Konfiguration, zur Ausführung geplante Kommandos, Update-Historie und Lizenzen. Der Scout Server wird i. d .R. als Dienst realisiert und ist immer aktiv.

Die Scout Konsole ist die Schnittstelle, um Änderungen im Scout Server vorzunehmen. Diese Komponente kann auf demselben System oder einem entfernten Rechner installiert werden. Aus Sicherheitsgründen sind zwei Sitzungen verfügbar. Einerseits der "Administrator", der die Scout Konfiguration ändern kann, und der "Gast", der als Helpdesk fungieren und Lizenzen eingeben kann. Die Scout Konsole kann auf mehreren Systemen installiert werden.

Nicht jede Software kann auf jedem Gerät installiert werden, da sie kompatibel mit eLux sein muss. Deshalb gibt es sogenannte Container, die eine Sammlung von Software darstellen, die auf dem Client installiert wird. Sie ist hardwareabhängig, es gibt daher eigene Container für jede Hardwareplattform.

ELIAS NG steht für "eLux Image Administration Service Next Generation" und ist frei verfügbar für alle Nutzer von Scout. Mit diesem Tool kann der Administrator die auf dem Client installierte Software exakt den Bedürfnissen des Anwenders anpassen. Diese Zusam-

---

[7] Technisch ist das nicht korrekt, da es schon Unterschiede zwischen lokalen Anwendungen und fernen Anwendungen gibt. Es stellt sich aber für den Anwender so dar.

[8] Hinzufügen von mehreren Administratoren mit unterschiedlichen Berechtigungen

menstellung wird als Image bezeichnet und in dem sogenannten "Image Definition File" mit der Dateiendung *idf* gespeichert. [Sco06]

4.1 Organisatorische Möglichkeiten

Organisatorische Maßnahmen hängen in der IT immer eng mit technischen Änderungen zusammen und sind streng genommen nur in seltenen Fällen isoliert zu betrachten. Die hier getroffene Unterscheidung ist nicht als absolut anzusehen. Es ist durchaus denkbar, organisatorische und technische Komponenten gegeneinander zu tauschen oder beiden Bereichen zuzuordnen.

Folgende organisatorische Komponenten sind Gegenstand der Betrachtung und sollen unter dem Blickwinkel der Optimierung der vollautomatischen Verteilung von Betriebssystemen näher erläutert werden:

- Standardisierung
- Zentralisierung
- ZBV-Arbeitsplätze (Zur Besonderen Verwendung)
- Erstellen von Konventionen

Im Gegensatz zu Normen kann ein Standard von einem Unternehmen unter Ausschluss der Öffentlichkeit entwickelt werden und dient der internen Durchsetzung von einheitlichen Verfahrensweisen mit dem Ziel der Austauschbarkeit von Rechnerkomponenten, Vereinfachung des Informationsaustausches und letztlich der Kostensenkung.

Eine Zentralisierung kann letztendlich ein Ergebnis konsequenter Anwendung von Standards sein mit dem Fernziel der Kostensenkung durch Einsparung dezentral vorgehaltener Infrastruktur.

Arbeitsplätze zur besonderen Verwendung (ZBV) sind ebenfalls ein Produkt von Standardisierung. Zur Unterscheidung gibt es letztlich Standard-Arbeitsplätze und ZBV-Arbeitsplätze, deren Anzahl durch die Bündelung von Sonderfunktionen minimiert werden soll. Letztere bilden umfangreichere Nutzungsmöglichkeiten, wie z.B. CD-Brenner, Scanner oder großformatige Farbdrucker und sollen möglichst sparsam eingesetzt und doch durch geschickte Verteilung einem großen Nutzerkreis zugänglich gemacht werden. Die Bestückung dieser ZBV-Arbeitsplätze mit Hard- und Software erfolgt dabei durch einen standardisierten internen Warenkorb, der alle Fälle und Einsatzmöglichkeiten abdeckt.

Nur durch die Einhaltung von Konventionen können alle Einsparpotentiale und der wirtschaftliche Einsatz von Ressourcen optimiert werden. Dazu zählt, dass gewisse Abläufe und Verabredungen eingehalten werden.

4.2 Technische Möglichkeiten

4.2.1 Methoden der Installation

Arbeitsplätze können im Prinzip auf zwei klassische und lange Jahre im Einsatz befindliche Methoden installiert werden. Die erste Methode setzt im ersten Schritt auf die möglichst automatisierte Installation eines Betriebssystems auf Zielrechnern, die durchaus unterschiedliche Hardware beinhalten können. Dazu wird eine sogenannte automatische Antwortdatei erstellt, die stellvertretend für den Installierenden alle im Laufe einer Installation auftauchenden Optionen "beantwortet", wie gefordert einstellt und somit Automatisierung ge-

währleistet. Man spricht in einem solchen Fall von unbeaufsichtigter Installation (engl. unattended setup). Erst im zweiten Schritt erfolgt die Ausbringung von Softwareanwendungen mit Hilfe von automatischer Softwareverteilung (engl. deployment), die aber nicht Bestandteil der unbeaufsichtigten Installation ist, sondern einen separaten zusätzlichen Prozess erfordert.

Abbilder (engl. images) auf Computer zu verteilen, ist die zweite weit verbreitete Methode der Rechnerinstallation. Hierbei werden in einem einzigen Schritt komplette Abbilder von Rechnern verteilt, die Betriebssystem und die benötigten Softwareanwendungen enthalten. In diesem Fall spricht man von einem dateibasierten Image. Es ist auch möglich, sektorbasierte Images zu erstellen. Die Größe dieser Images ist allerdings von den darauf enthaltenen Dateien unabhängig, und wird lediglich durch die Geometrie der physikalischen Festplatte bestimmt. Diese Imagetechnologie funktioniert i. d. R. nur auf Rechnern mit exakt identischer oder kompatibler Hardware.

Dateibasierte Images sind sehr schnell auf die Endgeräte verteilt, da sie einerseits komprimiert übertragen werden und andererseits nur die Dateien beinhalten, die auch von den installierten Softwarekomponenten tatsächlich benötigt werden. Im Gegensatz dazu werden bei automatischen Softwareinstallationen oft die gesamten Quelldateien auf den Rechner kopiert, obwohl im Laufe der Installation nur ein Bruchteil dessen wirklich benötigt wird. Somit sind kurze Installationszeiten gewährleistet. Allerdings "erkauft" man den Vorteil zu Ungunsten der eingeschränkten Hardwarekompatibilität der Images. Unbeaufsichtigte Installationen beschränken sich in den meisten Fällen auf das Betriebssystem, seine Komponenten und Updates. Unbeaufsichtigte Installationen sind bei weitem nicht für alle Softwareprodukte verfügbar und machen daher in der Folge oft eine (kostenpflichtige) professionelle Softwareverteilung notwendig. Inzwischen ist bei Vorhandensein eines Active Directory eine Softwareverteilung von Paketen im MSI-Format (Microsoft Installer) möglich. Allerdings gilt hier ähnliches wie vorher: Oft liegen die Installationsroutinen nicht im Microsoft Installer Format vor und müssen erst mithilfe von meist kostenpflichtiger Drittanbietersoftware[9] umgewandelt werden. Die Gesamtinstallation mit Softwareverteilung ist langsamer als die Imageinstallation, da hier die gesamte Software erst auf den Rechner übertragen werden muss, wo sie dann automatisch installiert werden muss. Liegen bei einer Abbildinstallation die Zeiten im Bereich von 10-20 Minuten im 100Mbit LAN, sind im gleichen LAN bei einer Softwareverteilung Installationszeiten von bis zu 90 Minuten und mehr keine Seltenheit. Die Vorteile der automatischen Softwareverteilung liegen aber wieder klar bei der späteren Verteilung von Softwareupdates oder neu hinzugekommener Applikationen. Hier kann mit reiner Imageinstallation nur wieder ein aktualisiertes aber komplettes Abbild verteilt werden. Aufwändig wird das Ganze bei Softwareinstallationen, die nur auf wenigen oder gar nur einem einzigen Rechner vorgenommen werden sollen. Dort muss quasi ein eigenes und separat zu pflegendes Image vorgehalten werden. Inzwischen kann man zwar mit sogenannten (Image-) Zuwachssicherungen arbeiten und den Aufwand damit reduzieren, aber in extrem inhomogenen Umgebungen kann ein Vorteil doch bei automatischer Softwareverteilung liegen. [Sym03]

Mischformen

[9] Für Windows 2000 hat Microsoft eine Lightvariante WINSTLE auf der CD-ROM mitgeliefert, s.a. auch kb257718 "How to create a third-party Microsoft Installer package"

Die Frage, welche Technik eingesetzt werden soll, muss im konkreten Fall jeweils neu beantwortet werden, dabei sind auch Mischformen von beiden Installationsarten denkbar.

Windows Vista wird in einer neuen Installationsvariante verteilt. Dabei wird ebenfalls zu Beginn ein Image verteilt, das im Microsoft eigenen Format WIM (Windows Image) vorliegt. Dieses wird, erst nachdem es auf dem Zielrechner ist, durch eine unbeaufsichtigte Installation angepasst. Gegenstand der Betrachtung ist aber Windows XP bzw. Windows Server 2003, wo diese Möglichkeiten nicht vorliegen. Wir werden aber in ähnlicher Arbeitsweise Windows XP und Server 2003 verteilen, allerdings nicht nur das Betriebssystem, sondern komplett installierte Rechner.

Application-Streaming

Ein weiterer Weg der Bereitstellung von Anwendungen wird seit einiger Zeit von der Firma Citrix mit dem sogenannten Anwendungs-Streaming (engl. "application streaming") beschritten. Diese Option steht seit dem Erscheinen der Enterprise Version 4.5 des Presentation Servers zur Verfügung. Beim Application-Streaming werden Desktop-Anwendungen "on demand" auf das Endgerät übertragen, wo sie in einer geschützten Umgebung lokal ausgeführt werden, ohne am Endgerät installiert werden zu müssen. Dieses "Pull-basierte" Bereitstellungsmodell ist eine Alternative zu lokalen Desktop-Installationen auf den einzelnen Endgeräten. Die gestreamten Applikationen laufen dort in einer isolierten Umgebung auf dem Endgerät. Damit werden von vornherein Versionskonflikte auf dem Endgerät vermieden, da die gestreamte Anwendung mit anderen installierten Applikationen nicht in Berührung kommt. Ähnlich arbeitet auch Microsoft Softgrid, wo Anwendungen ebenfalls in einer speziellen Laufzeitumgebung zum Einsatz kommen. Innerhalb dieser Laufzeitumgebung wird für die Anwendung ein komplettes Betriebssystem simuliert. Nachteile dieser Technik sind allerdings Anwendungen, die entweder eigene Gerätetreiber mitbringen oder direkt mit dem Betriebssystem kommunizieren müssen. Voraussetzung sind leistungsfähige Endgeräte und ausreichend Bandbreite und lokaler Speicherplatz, da diese gestreamten Pakete, die schnell eine Größe von einem Gigabyte und mehr annehmen können, auf den lokalen Clientrechner transferiert werden müssen. [LAN07]

Fazit

Auch wenn hier primär von der Verteilung von Microsoft Betriebssystemen die Rede ist, ist aber auch parallel dazu immer die Verteilung des Thin-Client Betriebssystems im Auge zu behalten. Auch dort geschieht das Deployment automatisch und mit minimalem manuellem Eingriff. Dort wird ebenfalls auf eine Lösung in Verbindung mit einem Web- und TFTP-Server gesetzt. Insofern dürfen die beiden Lösungen, Verteilung von Thin-Client und Fat-Client, nicht miteinander in Konkurrenz stehen, sondern sollen im Idealfall harmonieren.

4.2.2 Installationsvarianten von Microsoft Betriebssystemen

Die technischen Möglichkeiten, die von Seiten des Herstellers Microsoft zur Verfügung gestellt werden, gliedern sich in zwei Anwendungsbereiche: Die Neuinstallation oder die Updateinstallation.

Technik	Neuinstallation	Updateinstallation
manuelle Installation	X	X
automatisierte Installation (unattended)	X	X
RIS (Remoteinstallationsdienste)	X	
Sysprep (Systemvorbereitungsprogramm)	X	
SMS (Systems Management Server)		X

Tabelle 4.1 Übersicht der Programme für Neu- und Updateinstallation

Natürlich können manuell beide Varianten installiert werden, es interessieren aber nur die automatischen Optionen. Da die Rechner alle neu installiert werden, kommen prinzipiell noch drei verschiedene Möglichkeiten in Betracht. Der Systems Management Server scheidet aus, weil dieser eine vorhandene Installation mit installiertem SMS-Agenten voraussetzt.

Die automatische Installation mittels der Datei *unattend.txt* oder *winnt.sif* bietet komfortable Möglichkeiten die Installation zu steuern. Sie bietet erweiterte Optionen im Vergleich zu der Steuerdatei *sysprep.inf* des Systemvorbereitungsprogramms. Die Installation mittels *winnt.sif* setzt einen lokalen Datenträger mit der Windows CD voraus. Während des Startvorgangs der Betriebssysteminstallations-CD wird automatisch auf dem Diskettenlaufwerk nach der Datei *winnt.sif* gesucht. Wird sie gefunden, erfolgt die Installation anschließend automatisch anhand der gefundenen Informationen in dieser Datei. Wird die Datei *unattend.txt* genutzt, muss diese bereits beim Aufruf des Installationssetups als Parameter übergeben werden. Dies geschieht in der Form

```
winnt.exe /u:unattend.txt /s:source path
```

Dabei ist mit dem Parameter "/s" der Pfad zu den Quelldateien gemeint, die sich im Normalfall im Verzeichnis *i386* auf dem Datenträger befinden. Hier ist es aber möglich ein Netzwerklaufwerk anzugeben.

Mithilfe der Remoteinstallationsdienste können ein oder mehrere Abbilder eines Betriebssystems auf einem RIS-Server erstellt und gespeichert werden. Anschließend kann ein RIS-Abbild von einem Clientcomputer, der PXE unterstützt, über eine Netzwerkverbindung heruntergeladen werden. Die Installation des heruntergeladenen RIS-Abbilds kann vollständig automatisiert werden. Um RIS zu nutzen, muss die entsprechende Softwarekomponente RIS auf einem Distributionsserver installiert werden, der die Abbilder zur Verfügung stellt. Unterstützt der Zielcomputer kein PXE, ist alternativ ein Start mit einer speziellen Diskette möglich, die mit dem integrierten Tool *rbfg.exe* des RIS-Servers erstellt werden kann und Netzwerkkarten nach dem PCI-Standard unterstützt. [XPP02]

Alle vorgenanten Methoden haben den Nachteil, dass sie immer nur das Betriebssystem und keine zusätzliche Software installieren können. Des Weiteren ist eine Installation eines Betriebssystems auf Basis eines dateibasierten Images wesentlich schneller als eine Installation mit einem Datenträgermedium CD-ROM, sofern man eine Netzwerkanbindung des

Clients von 100 MBit/s unterstellt. Weiterhin ist eine unbeaufsichtigte Installation nur geringfügig schneller wie eine manuelle Installation, da der Vorgang in gleicher Weise abläuft, nur die Eingabe in einem Fall über die Tastatur und im anderen Fall über eine Datei geschieht.

4.2.3 Imagebasierte Installationen von Microsoft Betriebssystemen

Installationen, die imagebasiert sind, haben alle den gemeinsamen Nachteil, dass ein Image nur auf Computer übertragen werden kann, die die gleiche Hardwareabstraktionsschicht (engl. HAL = Hardware Abstraction Layer) benutzen. Benutzen Quell- und Zielcomputer unterschiedliche Hardwareabstraktionsschichten, wie beispielsweise Standard-PC-HAL und ACPI-PC-HAL, muss für jede Hardwareabstraktionsschicht ein eigenes Image erstellt werden. Unter Windows XP wird im Gerätemanager im Abschnitt Computer angezeigt, welche HAL derzeit verwendet wird. In diesem Beispiel die ACPI-Uniprozessor-HAL.

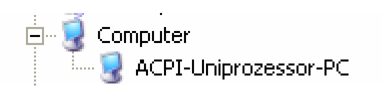


Abbildung 4.1 Anzeige der HAL im Gerätemanager

Server dagegen benutzen i. d. R. die ACPI-Multiprozessor-HAL, die die Unterstützung für mehrere Prozessoren bietet. Folgende Übersicht zeigt die Kompatibilitätsliste für die verschiedenen HAL-Versionen. Dabei steht die Abkürzung UP für Uniprozessor und MP für Multiprozessor.

Kompatibilität	ACPI PIC	ACPI–APIC UP	ACPI–APIC MP	Nicht-ACPI-UP PIC	Nicht-ACPI-APIC-UP	Nicht-ACPI-APIC-MP
ACPI PIC	X					
ACPI APIC UP		X	X			
ACPI APIC MP		X	X			
Nicht-ACPI-UP PIC				X		X
Nicht-ACPI APIC-UP				X	X	X
Nicht-ACPI APIC-MP					X	X

Tabelle 4.2 HAL-Kompatibilität

Dabei ist zu beachten, dass bei falscher Konfiguration zwar prinzipielle Funktionalität gewährleistet ist, aber unter Umständen mit Leistungsverlusten verbunden sein kann.

Die zwei möglichen Image-Techniken mittels RIS bzw. eines Drittanbieter-Tools Symantec Ghost werden im Folgenden genauer betrachtet. Die Techniken der unbeaufsichtigten Installation sind nicht die optimalen Maßnahmen, da dort nur das Betriebssystem automatisch installiert werden kann. Im Weiteren ist immer ein direkter Zugriff auf den zu installierenden Rechner notwendig, entweder in der Kombination CD-ROM und Diskettenlaufwerk oder das Booten mit einer netzwerkfähigen Diskette.

RIS ist eine kostenfreie integrierte Lösung von Microsoft und wird als optionale Komponente des Betriebssystems nachinstalliert. Ghost kommt deshalb in Betracht, da die KVMYK diese Software schon eingesetzt hat und Lizenzen in ausreichender Anzahl besitzt. Prinzi-

piell kann man jede andere Imagesoftware verwenden, sofern sie PXE und die Automatisierung über Befehlszeilenschalter unterstützt. Nur so ist eine Vollautomatisierung zu erreichen.

4.2.4 Installation mit Microsoft RIS

Im Rahmen der Installation der Remoteinstallationsdienste werden der BINL-Dienst (Boot Information Negotiation Layer), der Trivial File Transfer Protocol Daemon (TFTPD) und der Single Instance Storage Groveler (Einzelinstanzspeicherung oder SIS) installiert und ausgeführt. Der BINL-Dienst, die Hauptkomponente von Remote Installation Server (RIS), beantwortet PXE-Clientanfragen, validiert mithilfe des Active Directory den Client und übernimmt den Informationsaustausch zwischen Client und Server.
In dem dargestellten Beispiel sind nicht sämtliche Komponenten (RIS-Server mit BINL-Dienst, Einzelinstanzspeicherung und TFTP-Server) auf dem RIS-Server gespeichert, was aber unbedingt der Fall sein muss und nur aus Gründen der besseren Darstellung entzerrt worden ist. Es ist aber durchaus möglich, DHCP- und RIS-Server auf einer einzigen Maschine zu betreiben.
Die Voraussetzungen für Microsoft RIS sind:

- Server mit MS-Betriebssystem W2k-SP3 oder höher,
- Festplattenplatz in Abhängigkeit von Anzahl und Größe der "Images", aber separate Partition für i386-Quellen (XP, W2k3-Server, etc.) notwendig, die aber nur einfach abgelegt (Single-Instance-Store) werden,
- Microsoft DHCP-Server, der PXE-Clients unterstützt,
- Active Directory und DNS
- RIS-Software (TFTP-Daemon und RIS-Dienst = BINL-Dienst)

Der Prozess der Installation startet mit einem Netzwerkboot (PXE oder Startdiskette), in dessen Verlauf ein Clientinstallationsasssistent (CIW, Client Installation Wizard) heruntergeladen und durch die Taste "F12"[10] gestartet wird. Während der Ausführung des Assistenten muss sich der Benutzer in der Domäne authentifizieren, erst anschließend kann er das zu installierende Abbild auswählen.

[10] Nicht zu verwechseln mit dem Zeitpunkt an dem der PXE-Boot durch das Drücken der Taste F12 eingeleitet wird.

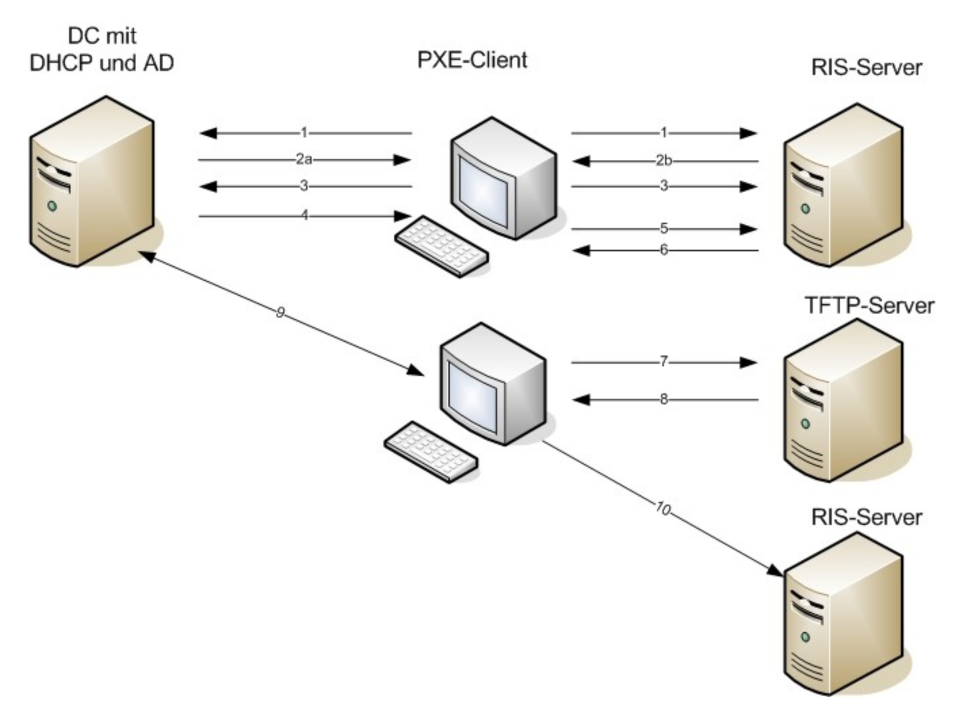


Abbildung 4.2 PXE-Boot Verlauf mit Microsoft RIS

1. Broadcast DHCP Discover
2. Broadcast DHCP Offer
3. Broadcast DHCP Request
4. Broadcast DHCP ACK
5. Broadcast DHCP Request (Option 66 anfragen)
6. DHCP-ACK
7. TFTP-Server (aus Option 66) anfragen
8. Datei (aus Option 67) von TFTP-Server laden
9. Authentifizierung im Active Directory
10. nach Authentifizierung TCP-Verbindung auf Port 4011 zum RIS-Server aufbauen

Der Client kann nur über PXE mit dem RIS-Server in Kontakt treten. Unterstützt der Client kein PXE, so ist es mittels einer speziellen Startdiskette, die PCI-Adapter unterstützt und mit *rbfg.exe* erstellt worden ist, möglich den Installationsprozess zu starten.
Der PXE-Bootverlauf ist im nachfolgenden erläutert. [Q244036] Nach dem Start setzt der Client ein DHCP-Discover (1) in Form eines Broadcasts ab, um TCP/IP-Konfigurationseinstellungen zu erhalten. Sowohl der DHCP-Server als auch der BINL-Server antworten auf den Broadcast (2a/b), allerdings unterscheiden sich die Antworten der beiden Server. Der DHCP-Server (2a) bietet nur eine IP-Adresse an, der BINL-Server (2b) bietet nur die Option 66 (TFTP-Server) an. Der DHCP-Client antwortet daraufhin mit einem DHCP-Request, um die angebotene IP-Adresse zu erhalten. Im Erfolgsfall antwortet der DHCP-Server dem Client mit einem DHCP ACK (4). Da der PXE-Client noch keinen TFTP-Servereintrag (DHCP-Option 66) erhalten hat, sendet er nochmals einen DHCP-Discover und die Vorgänge 1 bis 3 wiederholen sich in gleicher Form. Der PXE-Client antwortet diesmal aber dem RIS-Server mit einem DHCP-Request (5), da nur dieser die gewünschte Konfigurationsinformationen anbietet. Auch hier antwortet der RIS-Server mit einem DHCP ACK (6), in dem der gewünschte Startserver (DHCP-Option 66) und die Startdatei (DHCP-Option 67) übermittelt werden. Mit diesen Konfigurationsinformationen ausgestattet startet der PXE-

Client anschließend eine Anfrage an den TFTP-Server (7) und verarbeitet in der Folge die binäre Datei *startrom.com* (8). Ist die Datei ausgeführt, wird über die Taste "F12" der CIW gestartet und nach erfolgreicher Anmeldung (10) das Installationsmenü betreten.

Durch diese Konstellation, DHCP-Server und der "abgespeckte" DHCP-Server namens BINL, kommt es zu einem Problem mit der Verteilung der Thin-Client Betriebssystemsoftware eLux NG, wie man noch später sehen wird.

4.2.5 Imaging mit der Software Symantec Ghost

Die Symantec Ghost Solution Suite bietet mehrere Möglichkeiten der Installation. Es genügt aber die Option "Standard Tools installieren" auszuwählen. Diese Option installiert den "Ghost Boot Assistent", "Ghost Explorer" und den "GhostCast Server". Der "Ghost Boot Assistent" bietet die Möglichkeit Startdatenträger oder –disketten zu erstellen. Mit dem "Ghost Explorer" ist das Betrachten und nachträgliche Bearbeiten von erstellten Images eingeschränkt möglich. Der "GhostCast Server" ist, wie der Name vermuten lässt, der serverbasierte Teil der Imagetechnologie. Dorthin verbindet sich der Ghost-Client, um Images zu erstellen oder zurückzuspielen.

Der benötigte TFTP-Dienst kann mittels Freeware realisiert sein. Hier wird die OEM-Version der Software "3Com Boot Services" benutzt, die auf dem Datenträger von Symantec Ghost enthalten ist. Neben dem TFTP-Dienst wird u. a. noch der "Boot Image Editor" installiert, der es ermöglicht die TFTP-Startdatei (s. o. DHCP-Option 67) anzupassen. Das Besondere dabei ist, dass es damit möglich wird ein Menü zu hinterlegen, so dass man aus diversen Optionen auswählen kann. Wie das Menü aussieht und wie es erstellt wird, wird später im Kapitel 7.4 "Bereitstellung der Images über PXE" ersichtlich. Ein Beispiel ist im Kapitel 7.2.1 "Imageserver für Fat-Clients (Symantec Ghost)" dargestellt.

Die Voraussetzungen für Symantec Ghost Corporate Edition sind:

- Server mit MS-Betriebssystem W2k-SP3 oder höher,
- SQL-Server (MSDE oder höher),
- Festplattenplatz in Abhängigkeit von Anzahl und Größe der Images,
- vorhandener Standard DHCP-Server, der PXE-Clients unterstützt,
- vorhandener Standard TFTP-Server,

Der Ablauf ist dabei ähnlich dem des RIS-Servers aus dem vorhergehenden Kapitel.

1. Broadcast DHCP Discover
2. Broadcast DHCP Offer
3. Broadcast DHCP Request
4. Broadcast DHCP ACK
5. TFTP-Server (aus Option 66) anfragen
6. Datei (aus Option 67) von TFTP-Server laden

Der Unterschied liegt hier in der ersten Antwort des DHCP-Servers (2). Dieser bietet in einem Paket sowohl eine IP-Adresse als auch die Optionen 66 und 67 an. Der PXE-Client kann also hier nach erfolgreicher IP-Adressierung (4) direkt den TFTP-Server ansprechen (5) und die hinterlegte Datei (6) starten. Ist der Name der Datei beim RIS-Server fest vorgegeben, ist es in diesem Fall möglich die Datei in *pxelinux.0* umzubenennen. Dieser Name entspricht dem Namen, den der Scout NG Managementserver zur Verteilung der eLux NG Thin-Client Software benutzt.

4.2.6 Vor- und Nachteile der einzelnen Möglichkeiten

Lassen wir die unbeaufsichtigte Installation aus den vorgenannten Gründen außer Acht, bleiben nur zwei Varianten übrig: Die Installation mittels der Microsoft Remoteinstallationsdienste und die Nutzung der Imagetechnologie eines Drittherstellers, in diesem Fall Symantec Ghost.

Für die Remoteinstallationsdienste sprechen:

- Es ist eine kostenfreie Komponente des vorhandenen Betriebssystems.
- Die gute Active Directory Integration.
- Die Einarbeitungszeit ist gering und liefert schnelle Ergebnisse.
- Es ist leicht, die Installation an unterschiedliche Hardware anzupassen.

Nachteile von RIS sind:

- Die Installation muss zwingend auf Laufwerk C: erfolgen.
- Die Festplatte des Zielcomputers muss mindestens die gleiche Größe haben wie die des Referenzcomputers.
- Es kann nur das Betriebssystem installiert werden, keine Optionen oder sonstige Software.
- Es ist eine Benutzerauthentifizierung erforderlich.
- In Verbindung mit RIS ist die Nutzung eines zweiten TFTP-Servers zur Installation von Nicht-RIS-Clients nicht möglich, da die Kombination DHCP und BINL dies nicht zulässt.
- Es ist eine hohe Netzwerkbandbreite erforderlich, daher nicht für WAN-Strecken geeignet. Um Abbilder über Datenträger zu verteilen, ist eine Drittanbietersoftware (z.B. Ghost) notwendig.

Für den Einsatz von Symantec Ghost hingegen sprechen folgende Punkte:

- Die Festplatte des Zielcomputers kann auch kleiner sein, als die des Referenzcomputers. Sie muss nur so groß sein, dass alle Daten darauf Platz haben.
- Es können Abbilder fertig installierter Rechner verteilt werden.
- Es ist keine Benutzerauthentifizierung notwendig.
- Jeder DHCP-Server ist geeignet und der Einsatz von beliebig vielen TFTP-Servern ist denkbar.
- Abbilder können problemlos auf jede Art von Datenträger gebracht werden.
- Der Vorgang ist wesentlich schneller als alle anderen genannten Installationsmethoden.

Folgende Nachteile bringt der Einsatz von Ghost mit sich:

- Es ist eine zusätzliche kostenpflichtige Software.
- Die Einarbeitungszeit ist wesentlich höher als bei RIS oder unbeaufsichtigter Installation.
- Es kann schwierig werden, ein und dasselbe Image auf verschiedene Hardwareplattformen anzupassen.
- Es muss mit zusätzlichen betriebssystemspezifischen Zusatzprogrammen gearbeitet werden, z.B. *sysprep.exe* (Systemvorbereitungsprogramm von Microsoft)

Alleine aufgrund der Bedingung, dass auch die produkteigene Softwareverteilung von e-Lux NG auf PXE basiert und anschließend ein TFTP-Server kontaktiert wird, kann der Einsatz eines RIS-Servers fast ausgeschlossen werden. Es ist leider nicht möglich, den BINL-Server zu konfigurieren, d.h. irgendwelche nötigen Einstellungen zu verändern. So ist es

nicht möglich, anhand einer DHCP-Reservierung durch die Benutzung von unterschiedlichen Optionen (66 oder 67) Clients nach Typen (Thin- oder Fat-Client) zu unterscheiden. Es bliebe nur die Möglichkeit, die von Microsoft fest hinterlegte *startrom.com* so anzupassen, dass somit auch die Software eLux NG zu installieren ist. Diese Anpassung der *startrom.com* ist nur mit kostenpflichtigen Zusatztools möglich. Des Weiteren ist eine Authentifizierung notwendig, auch wenn kein Microsoftprodukt installiert werden soll, und anschließend muss noch manuell ausgewählt werden, ob ein Thin-Client oder Fat-Client Betriebssystem installiert werden soll. Alles das behindert die (voll-)automatische Installation und verhindert die Möglichkeit so früh wie möglich, nämlich bereits anhand einer DHCP-Reservierung, die Unterscheidung nach Thin- oder Fat-Client zu treffen.

Nach allen genannten Vor- und Nachteilen empfiehlt sich der Einsatz von Ghost, zumal der genannte Nachteil der kostenpflichtigen Beschaffung im vorliegenden Fall nicht besteht, da die Software bereits vorhanden ist und somit keine zusätzlichen Kosten anfallen.

4.3 Zusammenfassung der organisatorischen und technischen Probleme

Aus organisatorischen Gesichtspunkten sind die nachstehend aufgeführten Punkte besonders wichtig: Die angestrebte Lösung muss

- möglichst wirtschaftlich sein,
- einfach zu bedienen, warten und erweitern sein,
- an zentraler Stelle positioniert sein,
- flexibel sein,
- die Anzahl der ZBV-Arbeitsplätze minimieren,
- die Anzahl der Thin-Clients auf 75% erhöhen,
- die Störung im Betriebsablauf gering halten.

Auf technischer Ebene sind die folgenden Probleme zu lösen:

1. Problem: Es muss eine Lösung gefunden werden, wie sowohl Fat-Clients als auch Thin-Client Betriebssysteme automatisch verteilt werden.
2. Problem: Die Deploymentlösung soll nicht nur Betriebssysteme, sondern auch (einmalig) Anwendungen verteilen können.
3. Problem: Die Unterscheidung nach Thin-Client oder Fat-Client muss flexibel gestaltbar sein.
4. Problem: Die Unterscheidung nach Thin-Client oder Fat-Client muss zentral steuerbar sein.
5. Problem: Die Benutzereingriffe sollen möglichst minimiert werden.

Folgende Aufgaben sind ebenfalls zu bearbeiten:

1. Aufgabe: Die Verteilung auf die Rechner soll nach Möglichkeit über das Netzwerk, vorzugsweise per PXE-Boot, geschehen und Diskettenlaufwerke überflüssig machen.
2. Aufgabe: In Einzelfällen muss ein Diskettenlaufwerk unterstützt werden, z.B. für den Datenträgeraustausch (DTAUS).
3. Aufgabe: Es muss mit der angestrebten Lösung auch möglich sein, zur Datensicherung automatisch Images von Rechnern, vorwiegend ZBV-Arbeitsplätze und Server, zu erstellen.

4. Aufgabe: Es muss die Möglichkeit zum ausgiebigen Test außerhalb der produktiven Umgebung geschaffen werden.

Wünschenswert sind dabei die hier aufgelisteten Eigenschaften:

- Softwareverteilung für Nicht-MSI-Pakete oder sonstige administrative Wartungsaufgaben
- Konzept und Realisierung für die automatische Sicherung der Testanlage
- Automatische Übernahme und Aktualisierung der Active Directory Informationen von der produktiven Anlage ins Testsystem; evtl. mittels einer Drittanbietersoftware wie uMove von der Firma Algin Technology.

5 Lösungsansatz

5.1 Allgemein

Da fast alle Rechner neu beschafft werden und daher PXE-Boot beherrschen, ist bei diesen Rechnern kein Diskettenlaufwerk mehr vorgesehen. Im täglichen Umgang wird es kaum noch gebraucht, in seltenen Fällen ist noch der sogenannte Datenträgeraustausch (DTAUS) darüber realisiert. Sinnvoll ist daher die Einführung und Unterstützung von USB-Geräten (Diskettenlaufwerk, USB-Stick und Kartenlesegerät), die bei Bedarf gezielt eingesetzt werden können.

Da alle Endgeräte gleich sind, egal ob als Thin- oder Fat-Client eingesetzt, ist eine vergleichsweise niedrige Hardwareanforderung an die Geräte zu stellen. Unter der Prämisse der weiteren Forcierung der Terminalservertechnik kann davon ausgegangen werden, dass es nur noch in wenigen Fällen notwendig sein wird, lokale Anwendungen zu installieren. Wegen der dualen Einsatzmöglichkeit müssen die Geräte die Anforderungen von zwei Zertifizierungen erfüllen. Einerseits die Zertifizierung nach Microsoft-Richtlinien und andererseits die Anforderungen durch die Firma Unicon, Hersteller des Betriebssystems eLux NG. Server müssen ausschließlich den Microsoft-Zertifizierungsrichtlinien entsprechen.

Durch geschickte räumliche oder organisatorische Platzierung kann die Anzahl der ZBV-Arbeitsplätze gering gehalten. Diese können entweder nach organisatorischen (Gruppe, Referat, Abteilung) oder nach räumlichen Aspekten auf den Fluren verteilt werden.

Durch Einsparung der Server in den Außenstellen und konsequente Umsetzung der Terminalservertechnik für die Gesundheitsämter kann die Zentralisierung stärker ausgebaut werden. Vorteile ergeben sich durch Wegfall der dezentralen Verwaltung, Wartung, Pflege und Datensicherung der Server sowie den Einsatz von standardisierten Thin-Client Geräten, die im Fehlerfall innerhalb von wenigen Minuten vom Personal vor Ort ausgetauscht werden können. Durch Konsolidierung und Zentralisierung der Datenhaltung entfallen zukünftig auch aufwändige Replikationen der Datenbestände.

Durch eine sanfte Migration kann die Störung der Mitarbeiter während der Migration gering gehalten werden. Sanfte Migration bedeutet, dass die bestehende Windows NT Domäne und die neue Active Directory Umgebung parallel betrieben werden. Eine Migration der Benutzer, also die Überführung der Benutzerkonten von der alten auf die neue Umgebung, kann so z.B. referats- oder abteilungsweise durchgeführt werden.

Durch Festlegung der Bedürfnisse eines Standardarbeitsplatzes kann die dafür notwendige Software und Hardware festgelegt werden. Davon abweichende Anforderungen sind gesondert darzulegen und bedienen sich im Bereich der Hardware eines Warenkorbs, aus dem Komponenten ausgewählt werden können. Softwareanforderungen müssen im Einzel-

fall geprüft werden. Dabei ist zu überprüfen, ob die gewünschten Ziele nicht mit vorhandenen Mitteln genauso effektiv erledigt werden können.

Aus Gründen der Zukunftssicherheit ist die Softwareverteilung erweiterbar und möglichst flexibel bzgl. der Hardware zu entwerfen. Dabei darf die Nachvollziehbarkeit durch eine ausreichende Dokumentation nicht vernachlässigt werden. Die Unterscheidung nach Thin- oder Fat-Client wird an einer einzigen zentralen Stelle getroffen. Dazu bietet sich DHCP an. Dort ist jedes Gerät eindeutig registriert und an zentraler Stelle einfach zu überwachen.

5.2 Installationen von Clients und Servern

Die nachfolgende Prinzipzeichnung verdeutlicht dabei den grundsätzlichen Ablauf einer automatischen Installation von Rechnern. Dabei ist der Zielcomputer entweder ein Endgerät, also Thin- oder Fat-Client, oder ein Server.

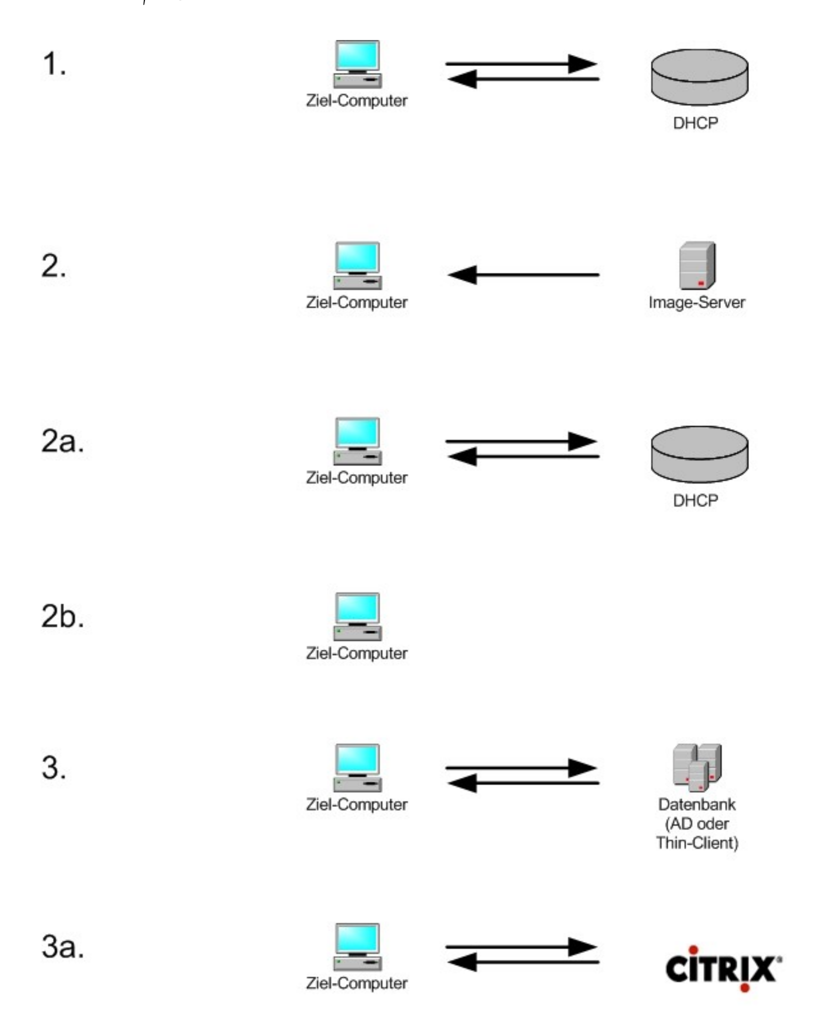


Abbildung 5.1 Prinzipieller Vorgang des Deployments

1. Ziel-Computer bootet per PXE und erhält von einem DHCP-Server eine IP-Adresse und Informationen über einen TFTP-Server.
2. Ziel-Computer erhält ein Image von einem Imageserver und startet neu.

2a. Ziel-Computer bootet und erhält von einem DHCP-Server eine IP-Adresse.

2b. Ziel-Computer ordnet sich anhand eines definierten Vorgangs einen NetBIOS-Namen zu und startet neu.
3. Ziel-Computer bootet erneut und schließt sich einer Active Directory Domäne (Fat-Client oder Server) an oder wird einer OU in der Thin-Client Datenbank zugeordnet.
3a. Ziel-Computer führt einmalig weitere administrative Tasks aus und schließt sich beispielsweise einer Citrix-Farm an.

Für Thin-Clients, die mit eLux NG installiert werden, gilt der oben genannte Ablauf in ähnlicher Weise und stellt sich im Einzelnen wie folgt dar:

1. Ziel-Computer bootet per PXE und erhält von einem DHCP-Server eine IP-Adresse und Informationen über einen TFTP-Server.
2. Ziel-Computer erhält ein Image von einem Imageserver, das auf einem "Image Definition File" basiert und startet neu.
3. Ziel-Computer bootet erneut und wird einer OU in der Thin-Client Datenbank zugeordnet und startet ggf. erneut.

Die Schritte 2a., 2b. und 3a. entfallen komplett.

5.3 Sicherung von Clients und Servern

Damit vorhandene Rechner, mit komplexen und vor allen Dingen aufwändigen Installationen, nach deren Fertigstellung einfach gesichert werden können, ist es notwendig Abbilder zu erstellen. Dies ist mit einer Imagesoftware, mit der auch Rechner installiert werden, möglich. In der KVMYK wird derzeit Symantec Ghost 8 eingesetzt. In Verbindung mit dem Systemvorbereitungsprogramm (*sysprep.exe*) von Microsoft und einem Script oder einem Programm, das automatisch Rechnernamen generiert, ist es so möglich, die beiden Vorgänge - das Erstellen und Zurückspielen eines Images - komplett zu automatisieren. Das Programm zur Bildung der Rechnernamen wird im Rahmen der vorliegenden Diplomarbeit entwickelt.

5.4 Testanlage

Die zwei vorhandenen Rechner vom TYP FSC Celsius, die im Vorfeld der Migration bereits als Testanlage fungiert haben, können zur Erstellung einer neuen Testanlage genutzt werden. Mit geringer Hardwareausrüstung kann so eine Testumgebung auf Basis des kostenfrei verfügbaren VMware Servers erstellt werden. Für einfache Tests kann der VMware Server genutzt werden, um darauf zum einen virtuelle Terminalserver und zum anderen virtuelle Test-PCs für Administratoren zu erstellen. Dazu kann je ein virtualisiertes Image genutzt werden, das im Bedarfsfall genauso schnell in Betrieb genommen werden kann wie ein reales Abbild. Auch hier funktioniert das Deployment grundsätzlich genauso, wie in Abbildung 5.1 des vorhergehenden Kapitels dargestellt.

In den beiden folgenden Kapiteln werden die vorgenannten Lösungsansätze genauer ausgeführt, vorausgesetzt, dass das Thema der vorliegenden Arbeit betroffen ist. Sofern die Tätigkeiten außerhalb liegen, kann man davon ausgehen, dass diese entsprechend ausgearbeitet und umgesetzt worden sind. Dazu zählen der Warenkorb, die Optimierung der ZBV-Arbeitsplätze, die Einsparung der Außenstellenserver, die Festlegung eines Standardarbeitsplatzes, die Erstellung eines Zeitplans für den referats- oder abteilungsweisen Austausch der Arbeitsplätze und die Migration der Benutzer.

5.5 Organisatorische Problemlösung

Aus den Ausführungen des letzten Kapitels ergeben sich die folgenden Aufgaben mit organisatorischem Schwerpunkt:

- Erstellen eines organisatorischen Konzepts für Images inkl. Versionierung und Dokumentation.
- Entwerfen eines Namenskonzepts für alle Netzwerkgeräte.
- Entwerfen eines IP-Adressen-Konzepts für alle Netzwerkgeräte.
- Entwerfen eines Konzepts für die Zuordnung Name zu IP-Adresse (MAC-Adresse).

5.6 Technische Problemlösung

Die Ausführungen der letzten Seiten haben die Entscheidung für eine imagebasierte Lösung mittels Ghost deutlich hervortreten lassen. Die folgenden drei Faktoren sind dabei die technisch Entscheidenden:

1. Die Verwendung von Microsoft RIS erlaubt keine gleichzeitige Nutzung von zwei verschiedenen TFTP-Serverdiensten.
2. Mit RIS sind nur Betriebssysteminstallationen möglich.
3. Mit RIS ist keine Sicherung der installierten Rechner zu Zwecken des Backups möglich.

Darüber hinaus ist bei RIS eine Authentifizierung während der Installation notwendig und damit eine Vollautomatisierung nicht möglich. Zwar sind zwei TFTP-Serverdienste parallel zu betreiben, aber der entscheidende Punkt ist die zentrale Steuerung, wie der Zugriff auf den TFTP-Server erfolgt. Hier ist die DHCP-Option 66 (TFTP-Server) entscheidend. Sobald der RIS-Server installiert und aktiviert wird, wird im Hintergrund diese Option im DHCP-Server deaktiviert. "Im Hintergrund" heißt, dass man diese Option 66 konfigurieren kann, aber der DHCP-Server sendet diese Option nicht in seinen Paketen an den Client. Dies wird aber erst deutlich, wenn man einen Protokollanalysator im Netz aktiviert und die dort versendeten Pakete beobachtet.

Daraus ergeben sich die folgenden Aufgaben:

- Erstellen eines DHCP-Server-Konzepts.
- Definition und Installation der Imageserver.
- Erstellung eines Images auf Basis der Definition eines Standardarbeitsplatzes.
- Entwerfen eines Konzepts für den datenträgerlosen Netzwerkstartvorgang.
- Vereinfachung des Ausbringens von Images ohne administrative Berechtigungen und manuelle Eingaben (durch weitgehende Automatisierung).
- Umsetzung des Konzepts zur Zuordnung von Name zu IP-Adresse bzw. MAC-Adresse mittels einer zu erstellenden Softwarekomponente.
- Erweiterung des Imagevorgangs mit anschließender Softwareinstallation.
- Erstellen eines technischen Konzepts für die Reduzierung der Anzahl der Images.
- Planung und Aufbau einer virtuellen Testanlage.
- Übertragung der Imagetechnologie in eine virtualisierte Umgebung.
- Vorbereitung der Arbeitsplätze der Administratoren zur Nutzung der Virtualisierungsmöglichkeiten.

Neben den oben aufgezählten Tätigkeiten gibt es noch einige Voraussetzungen, die erfüllt sein müssen, damit das Gesamtkonzept nicht an Kleinigkeiten scheitert. Dazu zählt, dass

vorhandene Steckerleisten gegen solche ohne Schalter ausgetauscht werden. Dies ist wegen Wake On LAN notwendig. Mit Hilfe von Wake On LAN ist es möglich, die Rechnerstationen nachts gezielt zu "erwecken" und somit ein nächtliches Update von Softwarekomponenten ohne Störung der Benutzer durchzuführen. Dazu muss eine permanente Stromversorgung der Geräte gewährleistet sein. Näheres ist im Kapitel "Imageserver für Thin-Clients (Scout NG)" erläutert.

6 Organisatorische Arbeiten

Die exakte Dokumentation der Entstehung und Weiterpflege eines Abbilds ist extrem wichtig. Weiterhin muss immer die gleiche Verfahrensweise an den Tag gelegt werden, wenn es darum geht, Änderungen oder Updates an dem Image durchzuführen. Dort können kleinere Änderungen oder Fehler große Auswirkungen haben, da die Abbilder auf viele Rechner dupliziert werden. Eine Versionierung der Abbilder ist von großer Bedeutung, um im Fehlerfall wieder an einen genau definierten Ausgangspunkt zurückkehren zu können. Fehlt diese Versionierung und Dokumentation kann ein Fehler u. U. nicht beseitigt werden und bedeutet im Extremfall, dass das Abbild komplett neu entwickelt werden muss.

Ein konsistentes Konzept für die Verteilung der IP-Adressen sorgt dafür, dass einerseits keine Konflikte durch mehrfache Verwendung gleicher Adressen entstehen und andererseits ist es durch geschickte Kombination von IP-Adressen und Namenszusätzen möglich ein eingängiges Namenskonzept für Rechner zu entwickeln. Ein definierter Zusammenhang zwischen IP-Adresse und Rechnername kann auch im Falle des Troubleshootings extrem hilfreich sein, um einfach und schnell eine Fehlkonfiguration aufzudecken. Heißt der Rechner z.B. *PC60117* und soll die zugehörige IP-Adresse 172.16.60.117 haben, kann im Fehlerfall eine Fehlkonfiguration schnell erkannt werden, ohne dass aufwändig in Listen oder im DHCP nachgeschaut werden muss, wie die Einstellungen korrekt lauten müssten. Vorteilhaft ist weiterhin, dass Konflikte und Fehler in DNS und WINS Konfigurationen durch vorgenannte Abhängigkeiten sehr leicht erkannt werden können

Der Mensch neigt dazu Namen zu vergeben, die er sich gut merken kann. Heißen im kleinen heimischen Netzwerk die Rechner vielleicht Mars, Pluto und Saturn, ist diese Art der Namensgebung aufgrund der Vielzahl der Rechner in einer größeren Umgebung nicht möglich. Eine Möglichkeit besteht darin, die Rechner einfach mit der zugehörigen Seriennummer zu benennen. Nicht nur, dass man sich einen Rechnernamen *YKXR134982* kaum merken kann, es ist sogar möglich, dass Rechner gar keine eindeutige elektronische Seriennummer haben. Selbst wenn Rechner auf dem Gehäuse eine Seriennummer haben, ist damit noch lange nicht gewährleistet, dass diese elektronisch im BIOS des Motherboards gespeichert ist und damit durch Hilfsmittel ausgelesen werden kann. Dies ist eigentlich nur bei namhaften Herstellern der Fall. Ähnliche Überlegungen gelten auch für sonstige Geräte im Netzwerk, wie Printboxen, Multifunktionsgeräte oder Netzwerkdrucker. Auch im telefonischen Support ist es wesentlich leichter, vom Anwender die PCID korrekt übermittelt zu bekommen. *PC60117* kann der Anwender besser und fehlerfreier mitteilen, als beispielsweise *5764QGFT832BA*.

6.1 Konzept für Images inkl. Versionierung und Dokumentation

6.1.1 Imageerstellung

Die Images werden immer primär auf der Hardwareplattform entwickelt, die zahlenmäßig am stärksten vertreten ist. Für die Endgeräte ist das der FSC Esprimo E 5700, für die Server der HP Proliant DL360 G4. Alle zu installierenden produktiven Terminalserver laufen auf der identischen Hardwareplattform des DL360.

Hardwareplattform	Anzahl
FSC Esprimo E 5700	360
FSC Scenic L (D1386)	20
FSC Scenic P 300	8
FSC Scenic E 600	3
FSC Esprimo E 5915	23
FSC Lifebook E 8310	8

Tabelle 6.1 Anzahl Endgeräte pro Hardwareplattform

Die Anzahl ist ein wenig geringer als die Zahl in der Tabelle 3.2. Dies ist dadurch begründet, dass es einige Einzelexemplare gibt, vor allen Dingen Laptops, die nicht aufgeführt werden. Diese werden von Hand installiert, da es derzeit den Aufwand nicht lohnt, für ein einzelnes Gerät das Image zu erweitern.

Das Image basiert auf den Standardanwendungen, die gemäß dem Standard-Arbeitsplatz respektive dem Standard für Terminalserver definiert worden sind. Diese Standardanwendungen sind ebenfalls auf allen Terminalservern installiert. Zu einem Standardarbeitsplatz gehören ebenfalls bestimmte definierte Plugins oder Add-ons für den Internet Explorer sowie die Druckertreiber. Im Weiteren gehören dazu die TWAIN-Treiber, um die Unterstützung des Scanvorgangs auf Basis dieser Schnittstelle zu gewährleisten. Diese Funktionalität ist ebenfalls fester Bestandteil der Terminalserverinstallationen.

Für Fat-Clients wird zusätzlich eine Software, der Netsupport Manager von NSM, für die Fernwartung der Rechner auf das Image installiert. Für Server wird dieser Zugriff aus der Ferne über den integrierten Microsoft Remote Desktop Client abgewickelt, bei den Thin-Clients erfolgt der Zugriff über die sogenannte Spiegelung, die integraler Bestandteil des Citrix Presentation Servers ist.

Die folgenden Vorgehensweisen und Einstellungen sind immer zu beachten:

- Der Name des Basisrechners, auf dem das Image entwickelt wird, lautet *PC60xyz*.
- Installationen werden immer mit dem Account des lokalen Administrators des Rechners vorgenommen.
- Installiert wird nur von CD-ROM bzw. DVD-ROM und dem Netzwerklaufwerk Z:, das aber niemals permanent verbunden wird.
- Es wird kein USB-Gerät aufgesteckt und von dort installiert, da alle USB-Geräte permanente Einträge in der Registrierung von Windows hinterlassen. Eine Ausnahme bildet der im Hause eingesetzte Scanner mit USB-Anschluss.
- Das Profil des Benutzers "Default-Benutzer" wird niemals von Hand verändert.

- Es wird so wenig wie möglich installiert und nur das, was tatsächlich gebraucht wird.
- Alle gerätespezifischen Treiber werden lokal im Verzeichnis *c:\install* gespeichert. Auf dieses Verzeichnis erhalten die Benutzer nur lesenden Zugriff.
- In das Verzeichnis c:\PnPdrvrs kommen die zum Zeitpunkt der Hardwareerkennung (Plug'n'Play) benötigten Treiber.
- Es wird keine Auslagerungsdatei verwendet.
- Das Systemlaufwerk wird mit "SYSTEM", ein evtl. vorhandenes weiteres Laufwerk wird mit "Volume" bezeichnet.
- Die Hintergrundbilder der KVMYK werden in das Verzeichnis, das durch die Umgebungsvariable %SystemRoot% (hier: c:\windows) angegeben wird, kopiert.
- Die betriebssystemeigene Firewall bleibt deaktiviert und wird ggf. durch zentrale Richtlinien über das Active Directory gesteuert und von dort bei Bedarf aktiviert. Gleiches gilt für das Sicherheitscenter.
- Die Option "Automatische Updates" bleibt deaktiviert und wird ebenso durch Richtlinien zentral gesteuert.
- Die "Systemwiederherstellung" bleibt für alle Laufwerke deaktiviert.
- Es wird eine Umgebungsvariable namens "ImageFile" erstellt, die immer die aktuelle Version des Images enthält und beispielsweise folgenden Wert haben kann: V.23-11.12.2007
- Die Computerbeschreibung lautet "Basisimage – Klon" und muss derzeit noch manuell angepasst werden, wenn die Installation des Images erfolgreich durchlaufen wurde.
- Nicht benötigte Betriebssystemkomponenten, wie Windows Messenger, Outlook Express und der MSN Explorer werden deinstalliert.

Diese Maßnahmen dienen vor allen Dingen dazu, Folgefehler zu vermeiden und das Image konsistent zu halten. Gewisse Einstellungen können sich bei unbedachter Verwendung durchaus auf alle Benutzer auswirken, die später an dem Rechner arbeiten. Deshalb sollte generell so wenig wie möglich an den Default-Einstellungen geändert werden. Alle notwendigen Einstellungen sollen später über die Gruppenrichtlinien zentral verändert werden.
Mit der Systemwiederherstellung können Benutzer im Falle eines Problems ihre Computer in einen früheren Zustand wiederherstellen, ohne dass persönliche Dateien verloren gehen. Standardmäßig erstellt der Windows Installer bei jeder Installation einer Anwendung automatisch einen Systemwiederherstellungsprüfpunkt. Wiederherstellungspunkte sind gespeicherte Zustände des Computers. Da Probleme im Zusammenhang mit Virenscannern auftreten können und darüber hinaus Änderungen am System nur durch Administratoren ausgeführt und benötigte Systemdateien nicht versehentlich durch normale Benutzer gelöscht werden können, kann auf den Einsatz der Systemwiederherstellung zugunsten der besseren Performanz und dem eingesparten Speicherplatz verzichtet werden. Auch hier ist es später jederzeit möglich über Richtlinien diese Einstellung anzupassen.

Die folgenden Standardvorgänge bilden immer den Abschluss, bevor ein neues Abbild erstellt wird:

- Das Bereinigen des Cache des Windows-Dateiüberprüfungsprogramms.
- Das Löschen von Dokumentenverknüpfungen.
- Die Bereinigung des Internet Explorers.
- Der Start der Datenträgerbereinigung.
- Das manuelle Löschen nicht mehr benötigter Dateien.

- Start der Defragmentierung des Datenträgers.
- Start der Fehlerüberprüfung des Datenträgers.
- Das Trennen von evtl. verbundenen Netzwerklaufwerken.
- Die Freigabe der derzeit verwendeten IP-Adresse mit *ipconfig /release*.
- Mithilfe des Programms *sfc.exe* kann der Cache, in dem Windows Kopien geschützter Systemdateien speichert, bereinigt werden. Erreicht wird dies durch den Befehl *sfc.exe /purgecache*.

Dokumentenverknüpfungen, die Historie der zuletzt geöffneten Dateien, können über "Start-> Systemsteuerung-> Taskleiste und Startmenü-> Startmenü-> [Startmenü-anpassen-erweitert | Klassisches Startmenü-anpassen] mit dem Button "löschen" bzw. "Liste löschen" gelöscht werden. Auch das Entfernen des Hakens bei "Zuletzt verwendete Dokumente auflisten" verhindert nicht, dass im Hintergrund diese Verknüpfungen nicht mehr gespeichert werden, sie werden nur eben nicht mehr angezeigt. Gespeichert werden diese Einstellungen im Ordner "Recent" des aktuellen Benutzerprofils. Bei einem deutschen Betriebssystem wird er als "Zuletzt verwendete Dokumente" angezeigt.

Beim Internet Explorer werden die temporäreren Internetdateien samt Cookies und der Verlauf gelöscht. Die anschließende Datenträgerbereinigung des Laufwerks C: bietet bereits einige Möglichkeiten, weitere temporäre und nicht mehr benötigte Dateien zu löschen. Die Datenträgerbereinigung befindet sich unter "Zubehör/Systemprogramme".

Im Rahmen von Installationen entstehen immer temporäre Dateien, deren Bereinigung nicht immer von den vorgenannten Maßnahmen als solche erkannt und automatisch entfernt werden. Diese sind händisch zu löschen und das Auffinden solcher Dateien hängt ein wenig von der Erfahrung und dem Gespür des Installierenden ab. Gute Ansatzpunkte sind aber immer das Profil des Benutzers, sowie das Verzeichnis *c:\Windows\Temp* sowie das Verzeichnis, auf das durch die Umgebungsvariable %temp% bzw. %tmp% verwiesen wird.

Die Fragmentierung bewirkt, dass die Datenträger mehr Suchvorgänge ausführen müssen, und führt somit zum Sinken der Übertragungsrate und damit letztlich zur Beeinträchtigung der Datenträgerleistung. Wenn Dateien auf einem Datenträger sehr fragmentiert sind, nimmt die Leistung merklich ab, weil die Plattenköpfe verschiedene Spuren auf dem Datenträger ansteuern müssen, um alle Cluster der Datei zu finden. Dieses Problem wird mit einer gezielten Defragmentierung mithilfe des Befehlszeilenprogramms *defrag.exe* oder über das Snap-In *Dfrg.msc* beseitigt. Mit beiden Tools werden die Dateien, Ordner und Programme auf der Festplatte des Computers neu angeordnet und so die Performanz des Datenträgers optimiert. Da zum Zeitpunkt der Defragmentierung keine Auslagerungsdatei vorhanden ist, wird diese anschließend durch den Systemvorbereitungsprozess ohne Defragmentierung auf dem aufgeräumten Datenträger erstellt.

Die Freigabe der aktuell verwendeten IP-Adresse ist Pflicht, bevor das zu verteilende Image erstellt wird. Dies verhindert später potentielle Probleme bei gleichzeitiger Ausbringung des Images an mehrere Rechner, die u. U. kurzzeitig identische IP-Adressen verwenden wollen.

6.1.2 Namenskonventionen und Versionierung

Das fertige Image, also die Datei mit der Endung *gho*, sollte sich an folgenden Namenskonventionen halten:

- Das fertige und zur automatischen Verteilung freigegebene Abbild trägt den Namen *PC60xyzVersionsnummer_sysprep.gho* und wird Masterimage genannt.

- Das Abbild, das die Vorstufe zum Masterimage bildet, und sich nur durch die fehlende Automatisierung durch die Systemvorbereitungsdienste davon unterscheidet, wird Basisimage genannt und als *PC60xyzVersionsnummer.gho* gespeichert.
- Das Arbeitsimage, das mit zusätzlichen Treibern für Massenspeichergeräte erweitert und somit für den Transfer auf eine neue Hardwareplattform vorbereitet wird, soll mit *PC60xyzVersionsnummer_driver.gho* benannt werden.
- Das Arbeitsimage *PC60xyzVersionsnummer_driver.gho*, das auf die neue Hardwareplattform aufgespielt wurde, wird in der Versionsnummer inkrementiert, in den Factory-Modus versetzt und mit dem Namen *PC60xyzVersionsnummer+1_factory.gho* weitergeführt.

Der Factory-Modus (*sysprep.exe –factory*), bietet die Möglichkeit zusätzliche Treiber oder Anwendungen zu installieren, ohne dass der Systemvorbereitungsvorgang ausgeführt wird. Die Versionsnummer entspricht einer fortlaufenden Zahl, die nach Ermessen des Installierenden inkrementiert werden kann. Dabei soll nach folgenden Maßstäben vorgegangen werden:

- Bei Transformation auf eine neue Hardwareplattform wird die Versionsnummer immer inkrementiert.
- Bei jeglichen Änderungen an dem Image muss die Versionsnummer inkrementiert werden, um offensichtliche Änderungen am Abbild anzuzeigen.
- Änderungen an dem Image sollen nicht permanent vollzogen, sondern die Änderungen sollen gesammelt und in einer nachvollziehbaren Einheit geändert werden. Erst dann folgt die Inkrementierung der Version.

Die Versionsnummer findet ferner auch im Eintrag "OEMDuplicatorstring = PC60xyz23-Basisimage-Sysprep-111207" der Datei *sysprep.inf* Anwendung. Dieser Eintrag hat gleichfalls nur informativen Charakter.

Im Anschluss an den Systemvorbereitungsvorgang muss bei den Endgeräten noch ein Dienst namens "PCinst" installiert werden. Dieser Dienst ermöglicht es scriptgesteuert beliebige Anwendungen auf dem Rechner zu installieren, nachdem dieser Teil des Active Directory geworden ist.

Auf den Terminalservern ist dies nicht notwendig, da diese über die integrierte Softwareverteilung namens Citrix Installationmanager, die eine Komponente des Citrix Presentation Servers darstellt, mit Installationen versorgt werden können. Dort muss aber anstatt des Dienstes "PCinst" automatisch die Software Citrix Presentation Server 4.0 installiert werden.

6.1.3 Dokumentation

Die Dokumentation erstreckt sich auf zwei Dokumente. Einerseits die ausführliche Dokumentation, die fortlaufend in einem einzigen Dokument fortgeführt wird. Andererseits eine kurze stichwortartige Datei im Textformat, die dem Image beiliegt und so heißt, wie das Image, das es beschreibt. Die Dateiendung ist dabei *txt*. Heißt das Image *PC60xyz23_sysprep.gho* heißt die zugehörige Datei *PC60xyz23_sysprep.txt*. Sie enthält im Kopfteil Informationen zu den unterstützten Hardwareplattformen (A) und des Betriebssystems (B). Anschließend erfolgen die Änderungen, die sich zum Abbild des Vorgängers ergeben haben (C) und ggf. sonstige Änderungen, die das Image nur indirekt betreffen. Dies sind z.B. Änderungen im Active Directory, damit der Imagevorgang korrekt abläuft.

```
# PC60xyz23_sysprep.txt
# HT/20071211
```

```
#
A) Platforms
Platforms supported & tested:
FSC Esprimo E 5700 (D2234)
FSC Esprimo E 5915 (D2344)
FSC Lifebook E 8310 (k.A.)
FSC Celsius H 250 (k.A.)

Platforms supported, but not (!) tested:
FSC Scenic L (D1386)
FSC Scenic P 300 (D1561)
FSC Scenic E 600 (D1534)

B) OS & Software
-MS Windows XP Professional -German-
++ Service Pack 2 installed
++ MSN Explorer not installed
++ Windows Messenger not installed
++ Outlook Express not installed


C) Changes (v.23)
-platform support for Celsius H250
++ SATA AHCI in BIOS must be enabled!
-automatic install of service "PCinst"

D) Third-Party
-change script for epo-agent in domain,
++ because of interaction while automatic installation cause epo-agent to crash
++ if c:\install\sysprep\1login.cmd is found, epo-agent will not be installed
```

Obiges Beispiel steht für das Image *PC60xyz23_sysprep.gho*, das für Endgeräte, die unter (A) genannt sind, entwickelt worden ist. Das Betriebssystem ist Windows XP mit SP2 (B) und in der Version 23 (C) ist die Unterstützung für das Laptop FSC H 250 und der Dienst "PCinst" hinzugekommen. Bei der Installation auf das Laptop H 250 ist darauf zu achten, dass im BIOS der AHCI-Modus des SATA-Controllers aktiv sein muss. Für die Installation des ePO-Agenten[11] muss das Script im Active Directory (D) angepasst werden, da dieses zeitgleich mit der automatischen Installation läuft und durch den Neustart während des Imagevorgangs nur teilweise installiert wird und in der Folge zu einem Fehler bei der Installation der Antivirensoftware führt.

6.2 Namenskonzept für alle Netzwerkgeräte

Entsprechend der Anzahl der Netzwerkgeräte werden entsprechende Zahlenbereiche festgelegt, die für die laufende Nummerierung der Geräte verwendet werden können. Dabei soll ein Zusammenhang zwischen der IP-Adresse und dem Namen des Netzwerkgerätes erkennbar sein. Die Namen der Geräte dürfen nicht länger als 15 Zeichen sein. Somit können potentielle Probleme in Verbindung mit dem Active Directory bzw. NetBIOS vermieden werden. Es werden nur Buchstaben und Zahlen verwendet; Sonderzeichen sowie reservierte Wörter finden keine Verwendung. [KB909264]

[11] ePO= McAfee ePolicy Orchestrator, unterstützt die automatische Ausbringung und Konfiguration der Antivirensoftware von McAfee

Netzwerkgerätetyp	Standort (IP-Netz)	Anzahl
Endgeräte	Kreishaus	>255
Endgeräte	GA Koblenz	<255
Endgeräte	GA Mayen	<255
Endgeräte	GA Andernach	<255
Endgeräte	Kfz Zulassung Mayen	<255
Printboxen + Netzwerkdrucker	Kreishaus	<255
Printboxen + Netzwerkdrucker	Alle Außenstellen	<255
Multifunktionsgeräte	Kreishaus + alle Außenstellen	<255
Server	Kreishaus	<255

Tabelle 6.2 Anzahl Netzwerkgeräte

Die Verwendung einer dreistelligen Zahl, die dem letzten Oktett der IP-Adresse des Geräts entspricht, reicht in fast allen Fällen aus. Die Ausnahme bildet die Anzahl der Endgeräte im Kreishaus, die deutlich über 255, aber stark unter 500 liegt. Um einen eindeutigen Zusammenhang zwischen Netzwerkgerät und Typ des Gerätes zu bilden, empfiehlt es sich, ein sprechendes Kürzel als Namensanfang zu verwenden.

Netzwerkgerät	Namensbeginn
Endgerät	PC
Printbox + Netzwerkdrucker	PB
Multifunktionsgeräte	MFP
Server	je nach Typ (s.u.)
Server – Fileserver	FILESRV
Server – Printserver	PRINTSRV
Server – Domänencontroller	DC
Server – Mailserver	EXSRV
Server - Backupserver	BACKUPSRV
Server – Terminalserver	CTXSRV
Server – Datenbankserver	DBSRV

Tabelle 6.3 Namensbeginn der Netzwerkgeräte

Da Server ausschließlich im Kreishaus verwendet werden und die Zahl bei keinem Typ größer als 100, wohl aber größer als 10 werden kann, reicht hier zur eindeutigen Kennzeichnung der Zusatz einer zweistelligen Zahl, die bei Bedarf immer mit führenden Nullen aufgefüllt wird. Dazu ein paar Beispiele:

- FILESRV01
- DCKV02
- CTXSRV12

Fast alle Server befinden sich in einem fest zugeordneten IP-Bereich, z.B. 172.16.1.x, und es gibt nicht immer eine direkte Abhängigkeit des letzten Oktetts mit der laufenden Nummer des Servers. Dies hat historische Gründe und betrifft hier auch nicht die automatische Installation von Servern, da diese Server ausschließlich von Hand installiert werden. Eine Ausnahme bilden die Terminalserver, die in einer großen Anzahl vorhanden sind. Diese besitzen einen eigenen IP-Bereich 172.16.2.x und dort ist ein direkter Zusammenhang zwischen letztem Oktett und laufender Nummer gegeben, d.h. dass der Server CTXSRV03 die korrespondierende IP-Adresse 172.16.2.3 besitzt. Eine weitere Ausnahme bzgl. der Namensgebung bilden die Domänencontroller. Diese beginnen mit dem Kürzel DC, gefolgt von dem NetBIOS-Namen der Domäne, die sie kontrollieren. Die anschließende Nummer spiegelt das letzte Oktett wider, das auf 2 Stellen verkürzt ist. Die Domänencontroller jeder NetBIOS-Domäne besitzen getrennte IP-Bereiche.

Bei den restlichen Geräten sind dreistellige Zahlen erforderlich, die ebenfalls mit führenden Nullen gefüllt werden. Um aus dem Namen auch direkt eindeutig den Standort des Gerätes herauslesen zu können, wurden sogenannte Standortschlüssel eingeführt.

Schlüssel	Standort (IP-Netz)	IP-Bereich
50	Kreishaus	172.16.50.1-250
60	Kreishaus	172.16.60.1-250
58	GA Koblenz	172.18.58.1-250
57	GA Mayen	172.17.57.1-250
59	GA Andernach	172.19.59.1-250
55	Kfz Zulassung Mayen	172.15.55.1-250

Tabelle 6.4 Standortschlüssel Endgeräte

Schlüssel	Netzwerkgerät	Standort (IP-Netz)	IP-Bereich
26	Printbox	Kreishaus	172.16.26.1-250
28	Printbox	GA Koblenz	172.18.28-1-250
27	Printbox	GA Mayen	172.17.27.1-250
29	Printbox	GA Andernach	172.19.29.1-250
25	Printbox	Kfz Zulassung Mayen	172.15.25.1-250

Tabelle 6.5 Standortschlüssel Printboxen

Schlüssel	Netzwerkgerät	Standort (IP-Netz)	IP-Bereich
24	Netzwerkdrucker	Kreishaus	172.16.24.1-250
23	Multifunktionsgeräte	Kreishaus	172.16.23.1-250

Tabelle 6.6 Standortschlüssel Netzwerkdrucker und Multifunktionsgeräte

Netzwerkdrucker unterscheiden sich technisch von Printboxen nur dadurch, dass die Netzwerkkarte fest in den Drucker eingebaut ist, während Printboxen ans Netzwerk angeschlossen werden und den Drucker von dort mit einem Druckerkabel ansteuern. Da es in der

KVMYK nur sehr wenige Netzwerkdrucker mit integrierter Netzwerkkarte gibt, ist es sinnvoll, diese auch durch die Namensgebung als solche kenntlich zu machen.

Druck-, Scan- und Kopierfunktionalität sind in einem so genannten Multifunktionsgerät vereint. Damit ist es beispielsweise möglich, an zentraler Stelle einzuscannen und die Ausgabe per E-Mail an den Empfänger zu senden. Auch diese Geräte sind nur wenig verbreitet. Der Name setzt sich in solchen Fällen aus dem gerätespezifischen Suffix (PC, PB, MFP), dem Standortschlüssel und dem letzten Oktett der IP-Adresse zusammen. Einige Beispiele seien zur Verdeutlichung hier angeführt.

- PC50117 (Rechner im Kreishaus mit der IP-Adresse 172.16.50.117)
- PC57003 (Rechner im GA Mayen mit der IP-Adresse 172.17.57.3)
- PB29012 (Printbox im GA Andernach mit der IP-Adresse 172.19.29.12)
- MFP23004 (Multifunktionsgerät im Kreishaus mit der IP-Adresse 172.16.23.4)

6.3 IP-Adressen-Konzept für alle Netzwerkgeräte

In enger Anlehnung an das Namenskonzept ergibt sich daraus auch das Konzept der Verteilung der IP-Adressen. Aufgeführt sind hier nur die Teile, die die automatische Verteilung von Images auf Endgeräte und Terminalserver betreffen.

Standort	Gerätetyp	IP-Adressen	
		von	bis
Kreishaus	Server	172.16.1.1	172.16.1.254
Kreishaus	Terminalserver	172.16.2.1	172.16.2.254
Kreishaus	Endgeräte	172.16.50.1	172.20.50.250
Kreishaus	Endgeräte	172.16.60.1	172.20.60.250
Kfz Zulassung Mayen	Endgeräte	172.15.55.1	172.15.55.250
GA Mayen	Endgeräte	172.17.57.1	172.17.57.250
GA Koblenz	Endgeräte	172.18.58.1	172.28.58.250
GA Andernach	Endgeräte	172.19.59.1	172.19.59.250

Tabelle 6.7 Zuordnung der IP-Adressen

Bei den Endgeräten werden die IP-Bereiche nur bis 250 ausgenutzt. Die höheren Adressen sind für andere Zwecke - vor allen Dingen zum Test - reserviert und werden für den produktiven Einsatz nicht verwendet.

6.4 Zuordnung Name zu IP-Adresse bzw. MAC-Adresse

Hier sind prinzipiell zwei Varianten denkbar, auf die im Laufe der automatischen Installation zurückgegriffen werden könnte. Beide bedienen sich dabei streng genommen nicht der Zuordnung der IP-Adressen zu Rechnernamen, sondern genauer gesagt der Zuordnung von MAC-Adressen zu Rechnernamen. Die MAC-Adresse bezeichnet die weltweit eindeutige Netzwerkadresse des Netzwerkadapters. Eine MAC-Adresse besteht aus 48 Bit und wird in der Regel in hexadezimaler Schreibweise dargestellt, z.B. 00:0C:29:12:34:56. Somit ist es also

möglich, anhand der MAC-Adresse einen Rechner eindeutig im Netzwerk zu identifizieren. Ganz korrekt ist dies nur insofern, als dass damit die Netzwerkschnittstelle eindeutig identifiziert ist. Diese kann allerdings nur in einem einzigen Rechner zu gleicher Zeit eingebaut sein und kann damit zum Zeitpunkt der Installation eindeutig bezeichnet werden.

Die erste Variante bedient sich der strengen Zuordnung von IP-Adresse zu Rechnername oder besser gesagt der möglichen Ableitung des Rechnernamens aus der IP-Adresse. Bekommt ein Rechner die IP-Adresse 172.16.60.7 zugeordnet, dann lässt sich anhand des bekannten Namenskonzepts daraus der Rechnername *PC60007* ableiten. Zwingende Voraussetzung für diese Art der Namensfindung ist eine DHCP-Reservierung. Eine feste DHCP-Reservierung sorgt dafür, dass einer bestimmten MAC-Adresse immer genau die gleiche IP-Adresse zugeordnet wird. Solange also nur eine einzige Netzwerkkarte in dem Rechner vorhanden ist und diese nicht ausgewechselt wird, erhält der zugehörige Rechner immer die gleiche IP-Adresse und kann damit automatisch installiert werden. Ändert sich die MAC-Adresse, z.B. durch Tausch der Netzwerkkarte oder bei onboard-Komponenten durch Austausch der Hauptplatine des Rechners, muss die DHCP-Reservierung auf die neue MAC-Adresse angepasst werden.

Eine andere Möglichkeit besteht in der direkten Zuordnung der MAC-Adressen zu Rechnernamen. So erhält man eine flexiblere Möglichkeit der Gestaltung der Rechnernamen und wird komplett unabhängig von der tatsächlich zugeordneten IP-Adresse. Dies ist in solchen Fällen erforderlich, in denen die IP-Adresse keinen Einfluss auf den Rechnernamen haben soll oder darf. Sollen beispielsweise mit ein und demselben Image zwei Rechner namens *ServerProd* und *ServerBackup* installiert werden, besteht sicherlich keinerlei Möglichkeit aus IP-Adressen die vorgenannten Namen abzuleiten. Besteht allerdings eine Liste in der Form 'MAC-Adresse des Rechners'='Rechnername' und ist es möglich, während des Installationsprozesses auf diese Liste zuzugreifen, kann eine korrekte Namenszuordnung erfolgen. Es besteht so theoretisch völlige Freiheit in der Zuordnung der Rechnernamen. Zu beachten sind dabei allerdings die im vorangegangen Abschnitt erläuterten Einschränkungen bei der Vergabe des Namens. Auch hier wirkt sich eine Änderung der MAC-Adresse in gleicher Weise aus, wie bereits oben gesagt. Auch hier wäre es natürlich möglich, die Liste in der Form 'IP-Adresse des Rechners'='Rechnername' aufzubauen. Dies hätte den Vorteil, dass Änderungen an der MAC-Adresse nur im DHCP angepasst werden müssen.

7 Technische Arbeiten

7.1 Voraussetzungen

7.1.1 DHCP in den Außenstellen

Die folgenden Kapitel beschäftigen sich mit der technischen Umsetzung der organisatorischen Konzepte und den Voraussetzungen für die automatische Installation. Die ursprünglich geplante Lösung mittels eines einzigen zentralen DHCP-Servers konnte aus technischen Gründen nicht realisiert werden; Details dazu finden sich in den nachfolgenden Kapiteln. Somit muss auch das gewünschte Ziel, in den Außenstellen keinerlei Serverinfrastruktur zu haben, leicht korrigiert werden. Bei der technischen Auslegung der DHCP-Server in den Außenstellen wurde ein Schwerpunkt auf die Wartungsfreiheit und Ausfallsicherheit gelegt. Hierzu wurden festplattenlose Workstations mit Serveraufgaben betraut. Diese booten ein kostenfreies Linux-Betriebssystem und lesen ihre Konfiguration von einer Diskette ein. Die Workstations sind ausgemusterte alte Arbeitsrechner, die nur ein Disketten- und CD-ROM-Laufwerk, eine Netzwerkkarte und mindestens 256 MB RAM besitzen müssen. Eingabegeräte und Monitor sind überflüssig, da die Einsatzbereitschaft durch akustische Signale kommuniziert wird. Ein Aus- und wieder Einschalten ist mittels des Gehäuseschalters jederzeit ohne Datenverlust möglich. Ersatz-CD, Ersatz-Diskette und ein Ersatzrechner sorgen für Ausfallsicherheit und schnelle Wiederinbetriebnahme im Fehlerfall. Diese Fehlerkorrekturen können in Zusammenarbeit mit dem zentralen telefonischen IT-Support von den Arbeitskräften in den Außenstellen durchgeführt werden. Anpassungen an diesen Servern sind nur bei Änderung von MAC-Adressen notwendig und daher selten, können aber zentral erledigt werden. Eine Sicherung der jeweiligen Konfiguration ist jederzeit zur zentralen Datensicherung möglich, da die Konfigurationsinformationen in Form von Textdateien vorliegen und in der Summe eine Größe von 80 KB nicht überschreiten. Diese Server dienen zusätzlich als dezentrale Wake On LAN-Proxyserver.

7.1.2 Imageserver

Imageserver sind der zentrale Punkt, an den sich die Clients wenden, die Imagesoftware starten und das Abbild herunterladen. Realisiert sind hier zwei verschiede Imageserver. Einer verteilt Abbilder für Fat-Clients und Server, der andere für Thin-Clients. Dies können zwei getrennte physikalische Maschinen sein; es ist aber technisch kein Problem, beide Imageserver auf einer Maschine zu vereinen. Dazu ist lediglich notwendig, einer einzigen Maschine die IP-Adressen der jeweiligen Hostnamen der Imageserver zuzuordnen. Auf diesen beiden unterschiedlichen IP-Adressen lauschen jeweils die zwei zugehörigen TFTP-

Server und geben ihre jeweilige Startdatei namens *pxelinux.0* an den Client weiter. Dann startet die *pxelinux.0* auf dem Server *SCOUTNG* die Installation von eLux NG für die Thin-Clients und gleichnamige Datei auf dem Server *IMAGESRV* den Ghost-Prozess für die Fat-Clients.

7.1.3 PXE

Das Starten über PXE ist eine zentrale Komponente, die durch das Drücken der Taste F12 während des Startvorgangs eingeleitet wird. Natürlich muss diese Funktion im BIOS des Rechners aktiviert sein. Bei neueren Rechnern ist dies meist die Voreinstellung. Eine korrekte Konfiguration des BIOS ist sehr wichtig und das nachfolgende Kapitel weist auf einige entscheidende Stellen hin. Um Fehlkonfigurationen vorzubeugen, wurde sogar die BIOS-Konfiguration automatisiert über PXE zur Verfügung gestellt.

7.1.4 Automatische Namensgebung

Die Umsetzung der gewünschten Namenskonvention innerhalb der Automatisierung ist nur über ein Script realisierbar, das automatisch die geforderten Namen erstellt und die Systemvorbereitungsdienste durch Microsoft entsprechend konfiguriert. Die Realisierung des Scripts erfolgt mit der Programmiersprache Perl. Perl steht für "Practical Extraction and Report Language" und hat seine Stärken, wie der Name bereits vermuten lässt, in der Verarbeitung sowie Ein- und Ausgabe von Daten. Perl wird seit 1987 von Larry Wall entwickelt und die aktuelle Version ist 5.10.002, die Version 6.0 liegt im Beta-Stadium vor. Vorteile sind vor allem in der komfortablen und schnellen Stringverarbeitung zu sehen. [Wal97]
Hier wurde die frei verfügbare Version "ActicePerl 5.10 for Windows (x86)" von Active State verwendet. ActiveState bietet die Version ActivePerl ebenfalls in der 64-bit Version und für andere Plattformen an, darunter AIX, HP-UX, Solaris, MAC, Linux oder als Source Distribution.

7.1.5 BIOS-Einstellungen

Die BIOS-Einstellungen der Server werden nicht angepasst. Diese sind bei der Installation der Hardware alle auf den aktuellen Stand gebracht worden und arbeiten in den Werkseinstellungen des Herstellers.
Bei den Endgeräten ist im BIOS auf die folgenden Einstellungen zu achten: Dabei ist die sicherste Vorgehensweise das Laden der Default-Einstellungen ("Get Default Values") und das anschließende Überprüfen bzw. Ändern der Einstellungen:

Nr.	Menüpunkt	Einstellung	Default-Einstellung
1	Diskette A:	"1,4M" oder "none"	"1,4M"
2	Boot Options/Quiet-Boot:	"disabled"	"enabled"
3	PC Beep Volume:	"LOW"	"HIGH"
4	Boot Sequence:	"HDD, PXE", Rest Disabled	"CD, Diskette, HDD, PXE"

5	LAN Remote Boot:	"enabled"	"disabled" (nur bei D1386!)
6	Hyper Threading:	"enabled"	"disabled" (nur bei P 300)

Tabelle 7.1 BIOS-Einstellungen

Die Einstellung (1) muss unbedingt korrekt sein, sonst kommt es bei eLux NG zu Problemen beim Start von veröffentlichten Anwendungen! Der Terminalserverclient liest die Einstellungen über Geräte aus dem BIOS aus und versucht bei falscher Einstellung z.B. ein Diskettenlaufwerk zu finden, das tatsächlich nicht installiert ist.
Ein "disabled" wird mit der Leertaste erreicht, vor dem Bootdevice erscheint dann das Zeichen "!". Die Reihenfolge wird mit den Tasten "+" und "-" geändert.
Die Einstellung (5) ist bei dem Scenic L mit dem Motherboard D1386 im Default "disabled" und muss daher dort auf "enabled" geändert werden. Bei den anderen Geräten ist der Default "enabled". Die Einstellung (6) - Hyper Threading - ist nur bei dem Modell P 300 mit Pentium 4 möglich. Allerdings ist per Default der Wert "disabled", daher muss der Wert dort auf "enabled" geändert werden.
Alle Einstellungen können per Hand oder per PXE-Boot vorgenommen werden. Das Einstellen per PXE-Boot hat den Vorteil, dass ein evtl. notwendiges Update des BIOS automatisch ausgeführt wird. Dazu muss nur einer der u. g. entsprechenden Menüpunkte ausgewählt werden, siehe dazu auch das Kapitel "Bereitstellung der Images über PXE".

- "BIOS-Update und SET E5700 (D2234)",
- "BIOS-Update und SET E 600 (D1534)"
- "BIOS-Update und SET P 300 (D1561)"
- "BIOS-Update und SET Scenic L (D1386)"

Die getesteten und freigegebenen BIOS-Versionen sind:

Rechnertyp	Motherboard	Version	Datum
FSC Esprimo E 5700	D2234	Version 5.00.R1.19.D2234	07/09/05
FSC Scenic L	D1386	Version 4.06.R1.06.D1386	05/07/03
FSC Scenic E 600	D1534	Version 4.06.R1.06.D1534	05/04/04
FSC Scenic P 300	D1561	Version 5.00.R2.07.D1561.01	07/16/04
FSC Esprimo E 5915	D2344	Version 6.00.R1.11.D2344.A3	05/23/07
FSC Lifebook E 8310	-	Version 1.10	12/27/07
FSC Celsius H 250	-	Version 1.12	12/27/07

Tabelle 7.2 Getestete BIOS-Versionen

Die Bezeichnung in den Klammern, die mit einem großen "D" beginnt, identifiziert eindeutig das Motherboard des Endgerätes. Das automatische Einstellen des BIOS über PXE ist noch nicht für alle Typen von Endgeräten vorbereitet.

7.2 DHCP-Server allgemein

DHCP (Dynamic Host Configuration Protocol) dient dazu, Clients automatisch mit den erforderlichen TCP/IP Protokolleinstellungen zu konfigurieren. Dazu ist es nicht notwendig,

jeden Client einzeln zu konfigurieren, sondern der Client fordert die Informationen bei Bedarf von einem DHCP-Server an. Dazu sendet er einen Broadcast (DHCP-Discover) ins Netz und erhält von dem DHCP-Server zur Antwort ein Angebot (DHCP-Offer) . Dieses Angebot, in Form von einer IP-Adresse und sonstigen Konfigurationsoptionen, kann der Client (DHCP-Request) annehmen und erhält dann die Bestätigung (DHCP-Ack) seitens des DHCP-Servers, dass er diese Konfiguration benutzen darf. Die Nutzungsdauer (lease time) kann zeitlich begrenzt werden, so dass der Client in zeitlich genau definierten Abständen zur Erneuerung seiner Lease nachfragt, oder dauerhaft in Form einer Reservierung gestattet werden. Eine Reservierung bedeutet, dass eine MAC-Adresse fest einer IP-Konfiguration zugeordnet wird, so dass der Client immer die gleichen Konfigurationsinformationen bekommt. Dies ist beispielsweise für Server sinnvoll, die immer unter der gleichen IP-Adresse angesprochen werden sollen.
Hier werden Reservierungen dazu genutzt, jedem Client eine quasi statische, aber zentral administrierbare, IP-Adresse zuzuordnen. Im Weiteren wird über DHCP festgelegt, ob der Client bei der Installation eine Thin- oder Fat-Client Konfiguration erhält. Dazu wird in der Reservierung die DHCP-Option 66 genutzt, in der der Name des TFTP-Servers festgelegt wird, der beim Installationsvorgang kontaktiert wird. Weiterhin wird die Option 12 genutzt, um einen eindeutigen Hostnamen für den Client anzubieten.
Es ist möglich, im DHCP-Server mehrere Bereiche mit unterschiedlichen Konfigurationen zu verwenden. DHCP-Optionen können global oder auf Bereiche festgelegt werden. So gelten diese nur für einen Bereich oder entsprechend übergreifend für alle Bereiche. Globale Serveroptionen werden auf die darunter liegenden Bereiche vererbt.

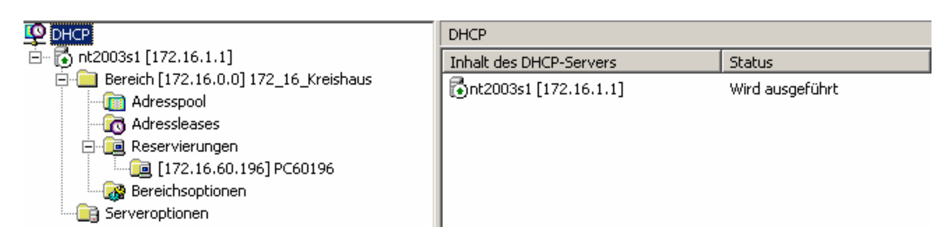


Abbildung 7.1 Übersicht Microsoft DHCP-Server

Da wir mindestens einen Bereich anlegen müssen und nur einen einzigen benötigen, legen wir auch die Optionen in diesem Bereich fest. Die Optionen, die für alle gelten und daher im Bereich festgelegt werden, sind:

Nr.	Beschreibung
003	Router
006	DNS-Server
015	DNS-Domänenname
044	WINS/NBNS-Server
067	Name der Startdatei

222	Scout NG Server
223	Scout NG Gruppen-ID

Tabelle 7.3 DHCP Optionen

Die beiden Optionen 222 und 223 sind herstellerspezifische Optionen und müssen im DHCP zusätzlich angelegt werden. Dazu wird eine Herstellerklasse "eLux NG" definiert.

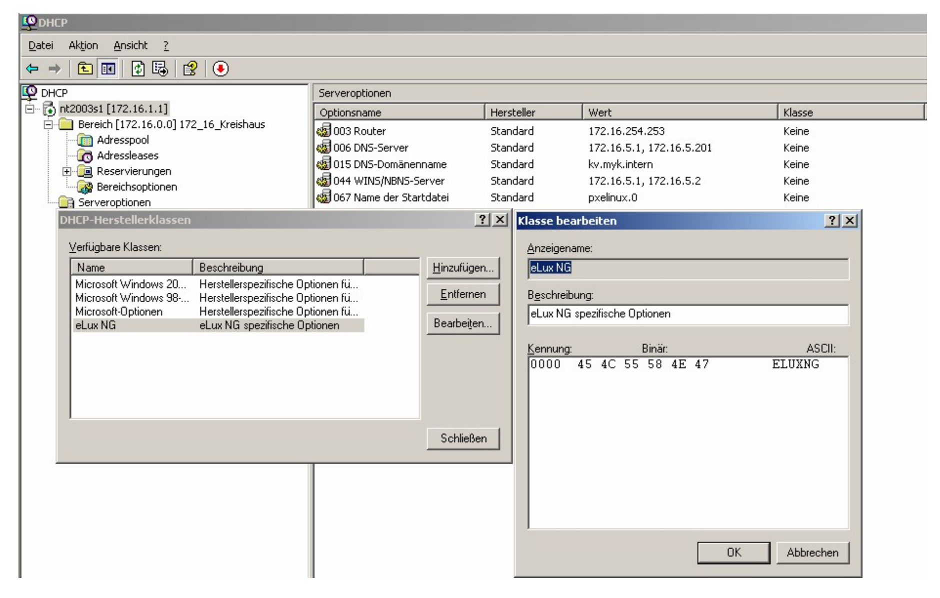


Abbildung 7.2 Herstellerklasse in DHCP

Die Herstellerklasse wird über den Menüpunkt "Herstellerklassen definieren" aus dem Kontextmenü des Servers aufgerufen. Als Name wird "eLux NG" gewählt, eine Beschreibung angegeben und im Kennungsfeld als Ascii-Code der Wert "ELUXNG" eingetragen. Nachdem die Klasse definiert ist, können ihr herstellerspezifische Optionen hinzugefügt werden. Dies geschieht durch "vordefinierte Optionen hinzufügen", ebenfalls aus dem Kontextmenü des Servers. Dort wird die eben erstellte Herstellerklasse "eLux NG" als Optionsklasse ausgewählt. Dann wird über den Button "Hinzufügen" eine Option mit dem Namen "Scout NG Server", dem Datentyp "Zeichenfolge", dem Code "222" und der Beschreibung "Name/IP des Scout NG Servers" erstellt.

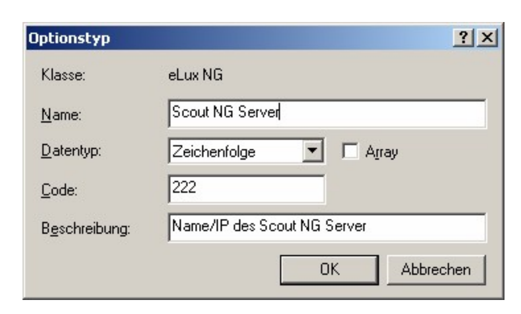


Abbildung 7.3 DHCP Optionstyp 222 "Scout NG Server"

Eine weitere Option wird mit dem Namen "Scout NG Gruppen-ID", dem Datentyp "Lang", dem Code "223" und der Beschreibung "Gerätegruppen ID Scout Enterprise Server" erstellt.



Abbildung 7.4 DHCP Optionstyp 223 "Scout NG Gruppen-ID"

Diese beiden herstellerspezifischen Optionen 222 und 223 werden von den Thin-Clients bei der Installation ausgelesen und unbedingt benötigt. Die genauere Bedeutung wird in Kapitel 7.3.2 erklärt.
Eine neue Reservierung für einen Fat-Client namens *PC60196* und der MAC-Adresse 00:11:22:33:44:55 würde dann beispielsweise folgendermaßen aussehen:

- 012: Hostname – Wert: *PC60196*
- 066: Hostname des Startservers – Wert: *IMAGSRV*

[172.16.60.196] PC60196

Optionsname	Hersteller	Wert	Klasse
012 Hostname	Standard	PC60196	Keine
066 Hostname des Startservers	Standard	IMAGESRV	Keine
003 Router	Standard	172.16.254.253	Keine
006 DNS-Server	Standard	172.16.5.1, 172.16.5.201	Keine
015 DNS-Domänenname	Standard	kv.myk.intern	Keine
044 WINS/NBNS-Server	Standard	172.16.5.1, 172.16.5.2	Keine
067 Name der Startdatei	Standard	pxelinux.0	Keine
222 Scout NG Server	eLux NG	SCOUTNG	Keine
223 Scout NG Gruppen-ID	eLux NG	0x2	Keine

Abbildung 7.5 DHCP-Optionen am Beispiel *PC60196*

Dabei sind die Optionen 12 und 66 pro Reservierung einzutragen, die anderen Optionen sind vom Bereich ererbt worden. Dabei kennzeichnet die Option 66 den Typ des Endgerätes, also Thin-Client oder Fat-Client. Die oben aufgeführten Optionen 222 und 223 ignoriert der Fat-Client, da er nicht darauf konfiguriert ist, die Herstellerklasse eLux NG zu benutzen.
Eine Reservierung eines entsprechenden Thin-Clients *PC60197* mit der MAC-Adresse 11:22:33:44:55:66 würde sich nur in den beiden Optionen 12 und 66 von der eines Fat-Clients unterscheiden:

- 012: Hostname – Wert: *PC60197*
- 066: Hostname des Startservers – Wert: *SCOUTNG*

Der Wert *Imagesrv* bzw. *SCOUTNG* steht dabei für den Namen des gleichlautenden Servers, der entweder den Ghostcast-Server oder den Installationsserver für eLux NG darstellt.

7.2.1 Standort: Kreishaus

Die Konfiguration des DHCP-Dienstes im Standort Kreishaus entspricht weitestgehend der Konfiguration aus den vorhergehenden Kapiteln und wird durch den Microsoft DHCP-Server-Dienst realisiert. Dieser Dienst läuft auf einem der Domänencontroller für die Domäne *kv.myk.intern* und beansprucht so keine eigene zusätzliche Maschine. Die Werte sind im Bereich namens "172_16_Kreishaus" wie folgt belegt:

Nr.	Beschreibung	Wert
003	Router	172.16.254.253
006	DNS-Server	172.16.1.1, 172.16.1.201
015	DNS-Domänenname	*kv.myk.intern*
044	WINS/NBNS-Server	172.16.1.1, 172.16.1.2
067	Name der Startdatei	*pxelinux.0*
222	Scout NG Server	*SCOUTNG*
223	Scout NG Gruppen-ID	0x1c

Tabelle 7.4 DHCP-Optionen Standort Kreishaus

Die Reservierungen, die hier getätigt werden, liegen in den IP-Bereichen:

- 172.16.1.x (Server)
- 172.16.2.x (Terminalserver)
- 172.16.7.x (Hostserver für virtuelle Server = VMware)
- 172.16.23.x (netzwerkfähige Multifunktionsgeräte)
- 172.16.24.x (Netzwerkdrucker)
- 172.16.26.x (Printboxen)
- 172.16.50.x (Endgeräte Kreishaus)
- 172.16.60.x (Endgeräte Kreishaus)

7.2.2 Standort: Außenstellen

Da auch in den Außenstellen Reservierungen benötigt werden, ist auch dort ein separater DHCP-Server notwendig. Bezüglich Broadcasts bildet ein Router eine natürliche Grenze und filtert diese aus, wobei es möglich ist, diesen Filtermechanismus[12] abzuschalten. Im Falle einer Anbindung über schmalbandige WAN-Strecken ist dies jedoch nicht empfehlenswert und führt zu einer unnötigen Belastung und damit zu einer Reduzierung der verfügbaren Bandbreite. Im Prinzip ist es durch Relay-Agenten möglich, in einer Außenstelle DHCP-Adressen zu vergeben. Dazu ist in einer Außenstelle kein eigener DHCP-Server erforderlich. Bei dieser Technik wird ein Gerät der Außenstelle, z.B. der Router vor Ort, so konfiguriert, dass er den lokalen Broadcast des DHCP-Clients entgegennimmt und an den dafür zuständigen DHCP-Server im Kreishaus weiterleitet (Relay). Entsprechend werden auch die Antworten des DHCP-Servers vom Relay-Agenten entgegen genommen und an den Client geleitet. Reservierungen sind bei dieser Technik nicht möglich, da der DHCP-Server immer "nur" ein und dieselbe MAC-Adresse des Relay-Agenten wahrnimmt und nicht die individuelle des Clients in der Außenstelle.

Vor dem Hintergrund der Ausfallsicherheit und der Ausnutzung vorhandener Ressourcen entschied man sich für eine Lösung mit geringer und wartungsarmer Hardwareanforderung. Die Wahl fiel auf die Open Source Software redWall-firewall[13], eine redhat-basierte Software-

[12] Ein entsprechend durchgeführter Test mit der Betreibergesellschaft T-Systems für die Router und WAN-Strecken musste abgebrochen werden, da es von Seiten der T-Systems nicht möglich war, die DHCP-Broadcasts über die Cisco Router zu transportieren.

[13] *http://www.redwall-firewall.com*

Firewall. Die Software bietet die Möglichkeit einer festplattenlosen Installation. Dabei wird die Software von CD gebootet und die Konfiguration auf einer Diskette abgespeichert und von dort während des Bootvorgangs eingelesen. Altrechner mit folgender Hardware sind ausreichend für den Betrieb eines DHCP-Servers:

- Rechnertyp FSC Scenic L Motherboard D1386
- P III 866 MHz
- 256 MB RAM
- CD-ROM
- PCI Netzwerkkarte 3Com 3C905TX

Die neben der onboard-Netzwerkkarte zusätzliche notwendige 3Com-Karte ist erforderlich, da die onboard-Karte unter dem Betriebssystem Linux kein Wake On LAN unterstützt. Dies ist der zweite Grund, in den Außenstellen einen Server zu installieren, da dieses Feature vor allen Dingen in der Verwaltungskonsole Scout NG für die Thin-Clients besonders wichtig ist. Aus Gründen der Ausfallsicherheit sind jedem DHCP-Server eine Ersatz-CD und eine schreibgeschützte Notfalldiskette mit der Konfiguration zum Zeitpunkt der Auslieferung beigefügt.

Der Betrieb des Servers ist auf das Fehlen jeglicher Ein- und Ausgabegeräte wie Tastatur, Maus und Bildschirm ausgelegt. Dazu ist im BIOS die Überprüfung dieser Geräte abgeschaltet und weiterhin nur das Booten von CD gestattet. Ist der Server mit dem Bootvorgang zu Ende und hat seine Konfiguration komplett eingelesen, signalisiert er seine Betriebsbereitschaft durch einen dreimaligen kurzen Piepton. Das Aus- und Einschalten des DHCP-Servers durch den Gehäuseschalter ist jederzeit ohne Datenverlust oder Beschädigung des Dateisystems möglich. Wird der Rechner im Fehlerfall mittels der Notfalldiskette gestartet, signalisiert er die Betriebsbereitschaft durch ein fünfmaliges langgezogenes Piepgeräusch und verdeutlicht damit den Notfallbetrieb. Da alle DHCP-Server die gleiche Hardware besitzen und sich lediglich durch die auf der Diskette enthaltene Konfiguration unterscheiden, ist derzeit ein zentraler Ersatzrechner im Kreishaus vorhanden, der im Notfall in die Außenstelle transportiert werden kann. Momentane Überlegungen tendieren zur lokalen Ersatzhardware in jeder Außenstelle, die auch von einem EDV-Laien auf einfachste Art und Weise in Betrieb genommen werden kann. Dazu wird der Ersatzrechner im Serverraum neben die produktive Maschine gestellt und im Fehlerfall muss lediglich die Diskette in den Ersatzrechner gesteckt und der Rechner angeschaltet werden. Dies entspricht einer einfachen, schnellen und transparenten Lösung, die in kürzester Zeit Betriebsbereitschaft garantiert.

Die Konfiguration des DHCP-Daemons[14] liegt als Datei *dhcpd.conf* im Ascii-Format vor und wird zentral vorgehalten. So kann im Notfallbetrieb durch einfaches Übertragen dieser Datei und den Neustart des DHCP-Daemons die reguläre Betriebsbereitschaft wiederhergestellt werden. Das Ändern und Übertragen der Konfigurationsdateien geschieht mittels SSH, einer verschlüsselten textbasierten Konsolensitzung, die beispielsweise mit dem Programm *putty*[15] und dem darin enthaltenen Hilfsprogramm zur Dateiübertragung *pscp* auch unter Windows möglich ist.

Die Werte sind im Bereich des Gesundheitsamtes Andernach wie folgt belegt:

[14] Unter Linux entspricht ein Daemon in etwa einem Dienst in der Microsoft Umgebung

[15] *http://www.chiark.greenend.org.uk/~sgtatham/putty/*

Nr.	Beschreibung	Wert
003	Router	172.19.254.253
006	DNS-Server	172.16.1.1, 172.16.1.201
015	DNS-Domänenname	*kv.myk.intern*
044	WINS/NBNS-Server	172.16.1.1, 172.16.1.2
067	Name der Startdatei	*pxelinux.0*
222	Scout NG Server	*SCOUTNG*
223	Scout NG Gruppen-ID	0x1d

Tabelle 7.5 DHCP-Optionen Standort Andernach

Die Reservierungen, die hier getätigt werden, liegen in den IP-Bereichen:

- 172.19.29.x (Printboxen)
- 172.19.59.x (Endgeräte)

In den übrigen Außenstellen sehen die Werte wie folgt aus, wobei nur noch die aufgelistet werden, die sich von den vorherigen unterscheiden.

Gesundheitsamt Koblenz:

- 003: Router – Wert: 172.18.254.253

Die Reservierungen, die hier getätigt werden, liegen in den IP-Bereichen:

- 172.18.28.x (Printboxen)
- 172.18.58.x (Endgeräte)

Gesundheitsamt Mayen:

- 003: Router – Wert: 172.17.254.253

Die Reservierungen, die hier getätigt werden, liegen in den IP-Bereichen:

- 172.17.27.x (Printboxen)
- 172.17.57.x(Endgeräte)

Kfz-Zulassung Mayen:

- 003: Router – Wert: 172.15.254.253

Die Reservierungen, die hier getätigt werden, liegen in den IP-Bereichen:

- 172.15.25.x (Printboxen)
- 172.15.55.x (Endgeräte)

7.3 Imageserver

In der KVMYK ist der Imageprozess als Client-Server-Lösung implementiert. Auf der Serverseite wartet der Ghostcast-Server auf Kontakt durch den Ghostcast-Client und legt das zu erstellende Windows-Image auf einem lokalen Datenträger des Servers ab. Dies ist über TCP/IP sowohl als Unicast als auch Multicast realisierbar. Bei den Thin-Clients wartet auf

der Serverseite der Scout NG-Prozess auf die Herstellung einer Verbindung durch den eLux NG-Client. Der Vorgang ist in Kapitel 7.4 in einer Grafik dargestellt.
Bei Thin-Clients und dem Betriebssystem eLux NG ist es nicht ganz korrekt mit dem Begriff Image zu arbeiten, zumindest nicht in dem Sinn, wie er bisher betrachtet worden ist. Hier wird kein Image erstellt, sondern es wird für die Thin-Clients eine Software-Konfiguration (≅ Image) festgelegt. Diese Software-Konfiguration bestimmt letztlich, ob überhaupt und welche Versionen der Software installiert werden. Also zum Beispiel wird der Citrix ICA-Client installiert (Ja oder Nein) und wenn ja, welche Version (8.1 oder 9.1). Diese Konfiguration ist für alle (!) Thin-Clients identisch, es wird also nur eine einzige Softwarekonfigurationsdatei benutzt. Insofern unterscheiden sich die einzelnen Thin-Clients nachher nur noch im Hostnamen und ihrer MAC-Adresse voneinander. Deshalb spricht man hier auch gerne von einem Image.
Die zwei vorhandenen Imageserver sind zwei getrennte Server, mit den Hostnamen *Imagesrv* und der IP-Adresse 1 und *SCOUTNG* mit der IP-Adresse 2. Beides sind Windows-Maschinen, da es sowohl die Scout NG-Verwaltungskonsole als auch die Software Symantec Ghost nicht für Open Source Betriebssysteme gibt. Dies ist zumindest im Fall der Scout NG Verwaltungskonsole ein wenig verwunderlich, da sie ein linux-basierendes Client-Betriebssystem managen. Anhand der eindeutigen DHCP-Reservierung für jeden Client wird anhand des Wertes der Option 66 entschieden, welcher der beiden Image-Server während des PXE-Bootvorgangs des Clients nachher angesprochen wird. Bei einem Fat-Client ist der Wert der Option 66 "*IMAGESRV*" und bei einem Thin-Client lautet der Wert "*SCOUTNG*". Über den TFTP-Serverdienst, der auf dem angesprochenen Server auf UDP-Port 69 lauscht, wird dann die Datei *pxelinux.0* an den Client übertragen und gestartet. Die Datei *pxelinux.0* enthält, je nach angesprochenem Server, einen unterschiedlichen Programmcode und wird in den nächsten zwei Kapiteln näher erläutert.

7.3.1 Imageserver für Fat-Clients (Symantec Ghost)

Die Serverkomponente besteht im Wesentlichen aus dem Symantec Ghostcast Server. Dieser wird über das Setup von Symantec Ghost mit diversen anderen Optionen, wie dem Ghost Explorer oder der Ghost Console, installiert. Der Ghostcast-Server wird als GUI[16] gestartet und ist mit wenigen Einstellungen sofort einsatzbereit.

[16] GUI = Graphical Benutzer Interface; grafisches Benutzerinterface

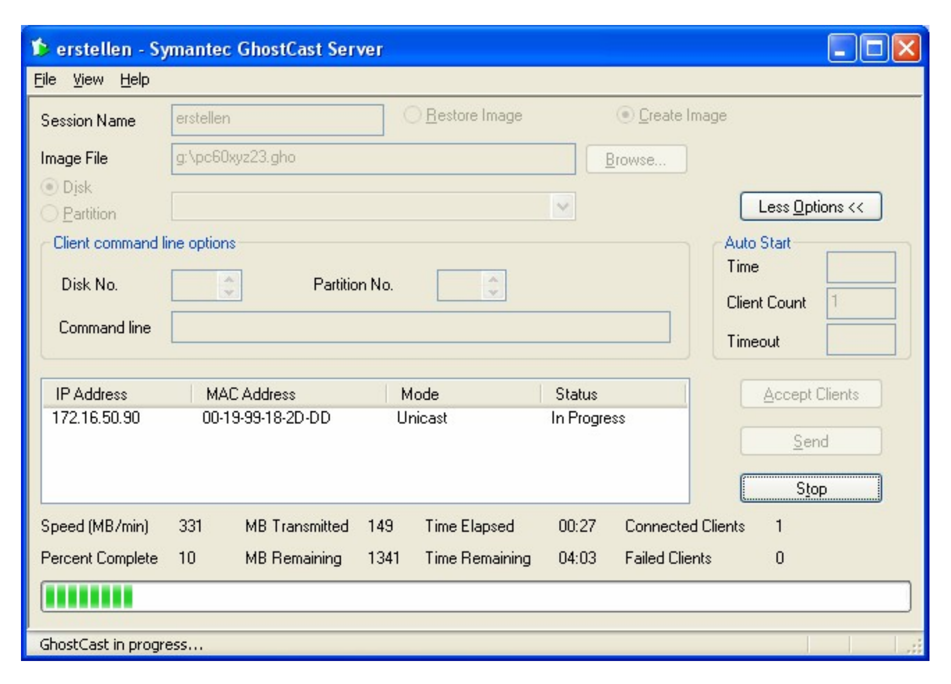


Abbildung 7.6 Symantec Ghostcast Server –Image erstellen

Die wichtigsten Einstellungen sind der Name der Sitzung (Session Name), die Auswahl Images erstellen oder Image zurückspielen (Create/Restore Image), ganze Festplatte oder nur eine Partition sichern (Disk/Partition) sowie der Zielpfad und Dateiname des zu erstellenden bzw. zurückzuspielenden Images (Image File). Wird der Server über "Accept Clients" gestartet, lauscht er auf TCP-Port 1106 und wartet auf den Client. Hat der Client Verbindung aufgenommen, ist er in der Liste mit IP-Adresse, MAC-Adresse und Mode (Unicast/Multicast) sowie aktuellem Status sichtbar. Im Falle des Erstellens eines Images startet der Server den Image-Vorgang direkt. Wird ein Abbild auf einen Rechner zurück gespielt, muss der Prozess zur Sicherheit mit "Send" bestätigt werden, um ein versehentliches Überschreiben des Zielrechners zu verhindern.

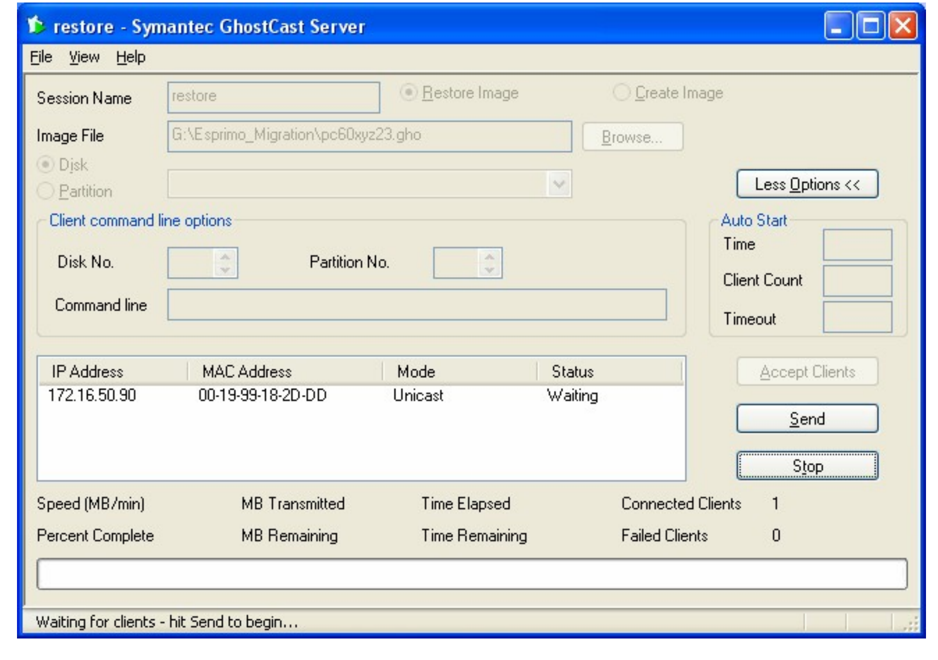


Abbildung 7.7 Symantec Ghostcast Server –Image Restore

Bevor die *ghost.exe* gestartet werden kann, hat der Client vom *Imagesrv* die Datei *pxelinux.0* geladen. Der Inhalt dieser Datei ist ein mit dem 3Com Boot Image Editor erstelltes Menü, welches einem gestattet, zwischen insgesamt 9 selbst erstellten Punkten auszuwählen. Der Default-Wert ist "Von lokaler Festplatte starten", der nach 5 Sekunden Zeitverzögerung ausgeführt wird.

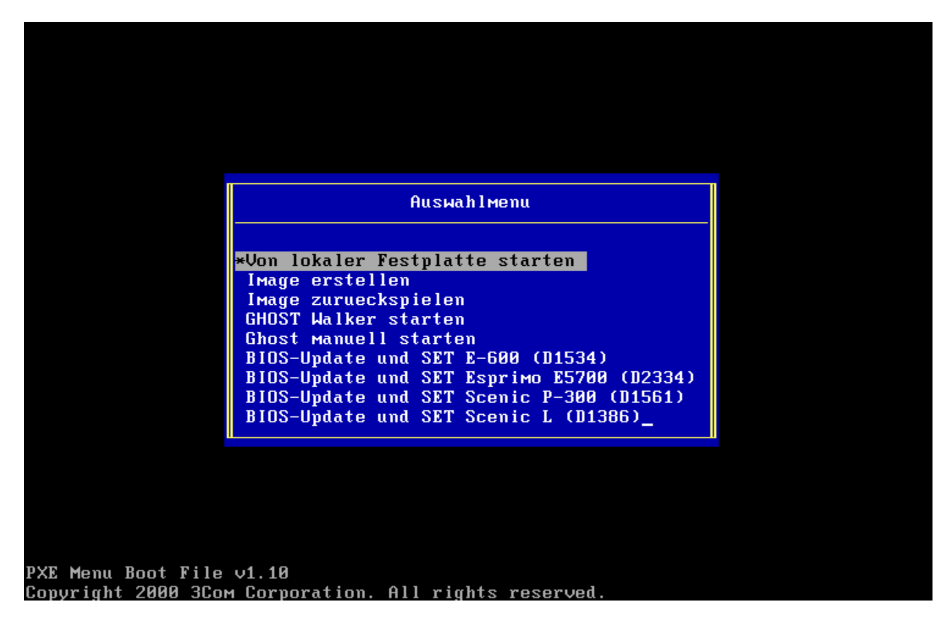


Abbildung 7.8 Menü nach PXE-Boot an Ghost Imageserver

Zum Erstellen eines Images wählt man den Menüpunkt "Image erstellen", zum Zurückspielen den Menüpunkt "Image zurueckspielen"[17] aus. Die restlichen Menüpunkte werden in Kapitel 7.4 näher erläutert. Hinter dem Aufruf von "Image erstellen" verbirgt sich letztlich eine kleine Imagedatei, die im RAW-Format[18] von einer Diskette erstellt wurde. Die Diskette wurde mit dem Symantec Ghost Boot Wizard erstellt, der Teil der Symantec Ghost Corporate Edition ist. Dieser Assistent bietet die Möglichkeit, diverse Bootdisketten oder Bootpartitionen zu erstellen. Zur Auswahl stehen:

- Standard Ghost Boot Package
- Network Boot Package
- Drive Mapping Boot Package
- CD/DVD Startup Boot Package with Ghost
- Console Boot Partition
- TCP/IP Network Boot Image
- TCP/IP Network Ghost Client Boot Image

Gewählt wurde hier das Network Boot Package, welches die Voraussetzungen erfüllt, um es über PXE zu booten. Die anderen Optionen haben entweder keine Netzwerkunterstützung, nur Netzwerkunterstützung für eine ganz bestimmte Netzwerkkarte oder sind einfach zu groß und werden damit nicht von PXE unterstützt und stürzen ab. Hinter dem Network Boot Package verbirgt sich eine abgespeckte DOS-Bootdiskette, die Netzwerkunterstützung (PXE Packet Driver *undipd.com*) bietet und den Ghost-Client automatisch startet. Die Parameter, die beim Aufruf der *ghost.exe* übergeben werden, sind noch auf die Bedürf-

[17] In diesem Menü ist die Darstellung von Umlauten nicht korrekt, daher "Image zurueckspielen"

[18] Erstellt mit rawritewin.exe Version 0.4 von http://www.chrysocome.net/rawwrite

nisse der KVMYK angepasst worden.
Siehe hier auch die Datei *autoexec.bat* dieser Startdiskette:

```
@echo off
SET TZ=GHO-01:00
prompt $p$g
\net\Undipd.com 0x60
MOUSE.COM
echo Lade...
CD GHOST
GHOST.EXE -clone,dst=@MCerstellen,src=1 -auto -RB -FX -QUIET -SURE -jm=u -Z2
```

Um sich an den Ghostcast Server zu verbinden startet der Client die *ghost.exe* bzw. *ghost32.exe*, je nachdem, ob es sich um ein 16-bit oder 32-bit Betriebssystem handelt.
Es ist möglich den Vorgang über eine GUI intuitiv zu starten, indem man über den Menüpunkt Ghostcast den Punkt Unicast auswählt. Anschließend bestätigt man die Einstellungen für den Ghostcast Server (automatisch) und den Sitzungsnamen sowie die Auswahl des Datenträgers und die Stufe der Komprimierung (nur beim Erstellen eines Images) und den eigentlichen Start des Vorgangs.

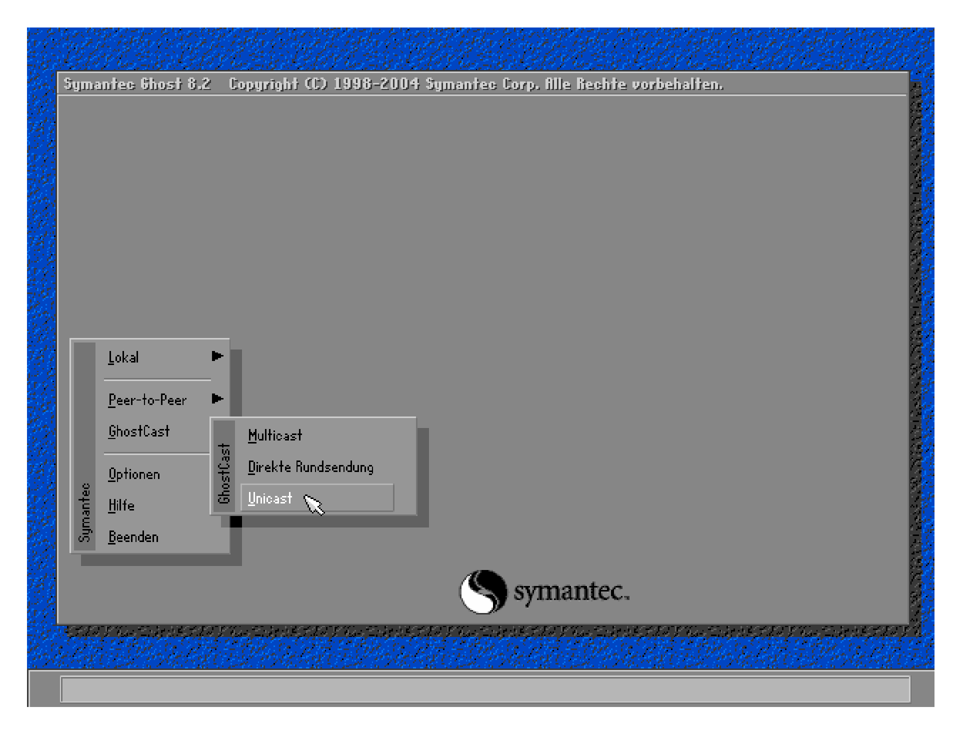


Abbildung 7.9 Symantec Ghost auf Clientseite

Nach Abschluss startet der Client-Rechner neu und der Ghostcast Server geht wieder in die Konfigurationseinstellung wie zu Beginn des Vorgangs. Alternativ kann auf der Clientseite alles in einer Befehlszeile eingegeben und somit komplett automatisiert werden. Dabei sieht die Befehlszeile beispielsweise folgendermaßen aus:

```
GHOST.EXE -clone,dst=@MCerstellen,src=1 -auto -RB -FX -QUIET -SURE -jm=u -Z2
```

Die Optionen bedeuten dabei:

-clone: Image erstellen
dst=@MCerstellen: Ziel ist die Sitzung mit dem Namen "erstellen"
src=1: die erste Festplatte im Computer
-auto: vergibt ohne Rückfrage Dateinamen, wenn das Image auf mehrere Dateien gesplittet wird

-*RB*: nach Beendigung den Computer neu starten
-*FX*: Programm wird nach Fertigstellung automatisch beendet
-*QUIET*: unterdrückt den Benutzereingriff
-*SURE*: übergeht alle sonstigen Meldung wie z.B. Partitionsgröße anpassen, etc.
-*jm=u*: Unicast benutzen
-*Z2*: Kompression "hoch"

Im Anhang 12.2 "Ghost Client am Beispiel 'restore'" ist der oben genannte Vorgang mit Printscreens dargestellt.

7.3.2 Imageserver für Thin-Clients (Scout NG)

Wie bereits schon angesprochen, definiert der Begriff Image beim Thin-Client Betriebssystem eine Softwarezusammenstellung. Diese Softwarezusammenstellung wird mit dem Programm ELIAS NG erledigt. Dabei wird ein sogenanntes Image Definition File mit der Dateiendung *idf* erzeugt. Diese *idf*-Datei wird dann über das Verwaltungsinstrument Scout NG den Clients zugeordnet. Der Imageserver namens *SCOUTNG* selbst besteht dabei aus folgenden Komponenten:

- Betriebssystem Microsoft Windows Server 2003
- UniCon Scout NG "eLux Manager - central management tool for eLux based ThinClients"
- UniCon ELIAS NG "eLux Image Administration Service - the image builder "
- TFTP-Server (Teil der Scout NG Installation)
- Apache Webserver (Teil der Scout NG Installation)

Im Rahmen der Installation wird der Dienst "Scout NG – Server" installiert, der automatisch beim Systemstart gestartet wird und auf den TCP-Ports 7777, 7778 und 22123 auf Verbindung wartet. Der ebenfalls mitinstallierte Scout-TFTP-Server erwartet eine Verbindung auf dem Standard TFTP-Port UDP 69 und gibt das lokale Verzeichnis *C:\Programme\UniCon\scoutng\tftpd* frei. Der Client lädt sich von dort die Datei *pxelinux.0* herunter. Die Datei *pxelinux.0* wird von der Software Scout NG als binäres Programm zur Verfügung gestellt. Das Programm *pxelinux.0* gibt die Meldung aus, deren Inhalt sich in der Datei *elux.msg* befindet, und startet anhand der Bootoptionen aus der Datei *elux.cfg* den Recovery-Kernel namens *eluxng.krn*. Dieser leitet eine Neuinstallation eines Thin-Clients ein und fordert zur Auswahl von A oder B aus. A steht dabei für die Installation von zertifizierten und getesteten Computern[19] und B bietet eine Installationsvariante für allgemeine PCs. Der Inhalt der beiden kleinen Dateien ist hier kurz dargestellt und ist weitestgehend typisch für das Bootverhalten von Linux.

```
elux.msg:

 eluxng.lss

 0fLoading  Recovery Kernel ......

elux.cfg:
DEFAULT elux
DISPLAY elux.msg
LABEL elux
KERNEL eluxng.krn
append initrd=eluxngdisk.gz root=/dev/ram0 video=vesa video=intel
```

[19] Eine Liste der zertifizierten Hardware befindet sich im Anhang oder online unter http://www.myelux.de/products.htm [Hinweis: kostenfreie Registrierung notwendig]

```
ramdisk_size=11264 idebus=33 console=tty04 quiet

ipappend 1
```

```
eLux NG - Recovery

ATTENTION: This installation will destroy all data on your harddisk (hda)!

Please select one of the available container:

  [ A ]  Platform specific container (UC_INTEL_P3)
  [ B ]  Generic PC container (UC_PC)

  [ Q ]  Quit installation
    a

  Format in progress, please wait ...
```

Abbildung 7.10 eLux NG Recovery-Installation[20]

Durch das Drücken der Taste "A" oder "B" wird der Installationsvorgang durch das Partitionieren der Festplatte gestartet und eine 5 GB große Startpartition erzeugt. Danach wird das angegebene Image Definition File ausgelesen und das dort spezifizierte Betriebssystem-Image durch einen Kopiervorgang auf die lokale Festplatte gebracht. Als Quelle wird dabei mit dem Benutzer *elux* und dem Kennwort *elux* auf den Apache-Webserver auf 172.16.10.140 (=IP-Adresse des Rechner *SCOUTNG*) zugegriffen, welcher in der Datei *eluxng.des* definiert wird und nachfolgenden Inhalt enthält: [eLu06]

```
[Recovery]
;PROXY=1.1.1.1
;PORT=80
URL=http://elux:elux@172.16.10.140/eluxng/__CONTAINER__/myk_V1_37-3.idf
;
```

In diesem Beispiel lautet das Image Definition File "*myk_V1_37-3.idf*" und /__CONTAINER__/ steht als Platzhalter für den eLux-Container und wird automatisch durch UC_Intel_P3 (= Auswahl "A") oder UC_PC (= Auswahl "B") ersetzt. Der Container ist der plattformspezifische Behälter mit den entsprechenden Softwareinstallationsdateien. Die Verzeichnisstruktur auf dem Apache Webserver ist im folgenden Printscreen dargestellt. Im oben genannten Beispiel wird nach Drücken der Taste "A" in das Unterverzeichnis *UC_INTEL_P3* verzweigt und die dort enthaltenen Programmdateien entsprechend dem Image "*myk_V1_37-3.idf*" kopiert. Das Wurzelverzeichnis des Webservers ist dabei "*C:\Programme\Apache Group\Apache\htdocs*".

[20] Nach Angaben des Herstellers wird es in zukünftigen Versionen möglich sein, diesen Vorgang komplett zu automatisieren, indem in der *elux.cfg* ein entsprechender Parameter hinterlegt und beim Aufruf übergeben wird.

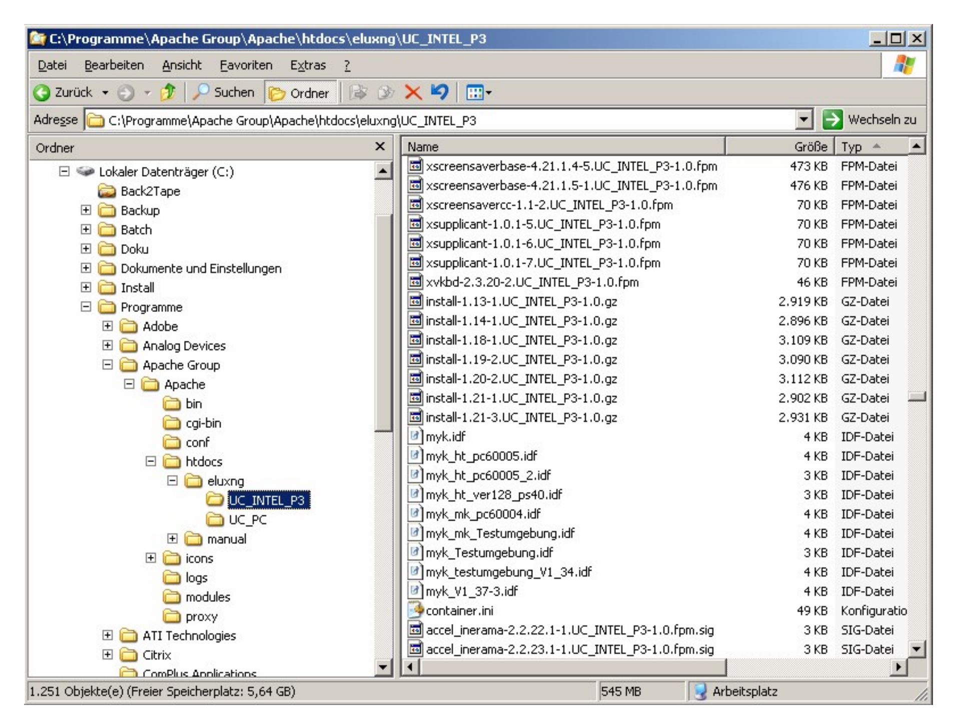


Abbildung 7.11 Apache Webserver mit Containern

Nach erfolgtem Kopiervorgang startet der Thin-Client automatisch erneut und ist sofort einsatzbereit. Der gesamte Vorgang vom ersten Start bis zum fertigen Thin-Client dauert ca. 5 Minuten, wobei der eigentliche Partitionierungs- und Kopiervorgang in einem 100 MBit Netzwerk nur ca. 90 Sekunden braucht. Entsprechend dauert dieser Vorgang in einer Außenstelle, die mit 2 MBit angebunden ist, knapp 7 Minuten, wobei sich im Wesentlichen nur der Kopiervorgang von 90 auf ca. 180 Sekunden verlängert.

7.3.3 Wake On LAN für Thin-Clients

Über die Verwaltungskonsole Scout NG ist es möglich, ein neues Image über Wake On LAN zu verteilen. Wake On LAN, oder auch oft nur kurz WOL genannt, kann dazu genutzt werden, einen ausgeschalteten PC zu starten. Dazu darf die Hauptplatine des PCs nicht komplett vom Strom getrennt sein und die Netzwerkkarte muss Verbindung zum aktiven Netz besitzen. Die Netzwerkkarte lauscht dabei auf ein für sie bestimmtes Datenpaket, das Magic Packet. [AMD95] Sobald sie dieses erhält, leitet sie einen Steuerimpuls an die Hauptplatine des Rechners, der den Rechner in der Folge anschaltet. So ist es möglich, gezielt einzelne oder eine ganze Gruppen von Rechnern anzuschalten, um beispielsweise zeitgesteuert nachts ein Update eines Images zu machen.

Dieses Magic Packet wird aber nicht über Router geleitet, so dass in den Außenstellen diese Technik nicht ohne weitere Hilfsmittel eingesetzt werden kann. Da in den Außenstellen ein DHCP-Server verwendet wird, kann dieser als Wake On LAN Server dienen. Dazu wird auf dem Server der Außenstelle ein Perl-Script hinterlegt, welches bei Bedarf von der Verwaltungskonsole aus per Batch-Script gestartet werden kann. Die Verzeichnisstruktur auf dem Server *SCOUTNG* sieht dabei so aus:

```
C:\Batch\Putty-Scripts\GaAn_Wol.cmd
C:\Batch\Putty-Scripts\GaKo_Wol.cmd
C:\Batch\Putty-Scripts\GaMy_Wol.cmd
C:\Batch\Putty-Scripts\plink.exe
C:\Batch\Putty-Scripts\GaAn\pc59001_wol.cmd
C:\Batch\Putty-Scripts\GaAn\pc59002_wol.cmd
```

```
…
C:\Batch\Putty-Scripts\GaAn\pc59008_wol.cmd
C:\Batch\Putty-Scripts\GaAn\wakeall.cmd
```

Der Inhalt des Scripts *pc59001_wol.cmd* besteht eigentlich nur aus einer einzigen Zeile:

```
..\GaAn_wol pc59001
```

Genauso wie das Script *wakeall.cmd*:

```
..\GaAn_wol wakeall
```

Es wird das Script *GaAn_wol.cmd* mit dem Parameter "pc59001" bzw. "wakeall" aus dem übergeordneten Verzeichnis aufgerufen, wobei das Script *GaAn_wol.cmd* so arbeitet:

```
@ECHO OFF
REM GaAn 172.19.0.0 mit DHCP-Server auf 172.19.1.1
SET redwall=172.19.1.1
IF %1%Test==Test GOTO NoPCName
..\plink -ssh -pw password -v -l root  %redwall% sh /etc/wakeonlan/%1%
```

Im Kern des Scripts wird die *plink.exe* mit diversen Parametern aufgerufen, die eine SSH-Sitzung auf 172.19.1.1 ausführt und dann dort lokal das Script */etc/wakeonlan/pc59001* bzw. */etc/wakeonlan/wakeall* aufruft. Die *plink.exe* ist Teil des Putty-Paketes und bietet die Möglichkeit *putty* über die Kommandozeile zu bedienen. Im Verzeichnis */etc/wakeonlan* auf dem DHCP-Server der Außenstelle liegen die über *plink* aufgerufenen Scripte.

```
\etc\wakeonlan\pc59001
\etc\wakeonlan\pc59001.mac
\etc\wakeonlan\pc59002
\etc\wakeonlan\pc59003.mac
…
\etc\wakeonlan\wakeall
\etc\wakeonlan\wakeonlan
```

Zu jedem Script gehört eine gleichnamige Datei mit der Endung *mac*. Diese enthält den Broadcast, der an das eigentliche Perl-Script *wakeonlan* als Parameter übergeben wird.
Inhalt des Scripts *pc59001*:

```
#! /bin/bash
WolHost=pc59001
ScriptPath=/etc/wakeonlan
MainPerlScript=wakeonlan
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$WolHost.mac
```

In der zugehörigen Datei *pc59001.mac* steht nur: `00:11:22:33:44:55 255.255.255.255`
Wird also eine Netzwerkkarte in einem Endgerät ausgetauscht, so braucht (neben dem DHCP-Server) nur in der entsprechenden Datei *Hostname.mac* die MAC-Adresse angepasst zu werden. Das Script *wakeall* ruft einfach alle vorhandenen Scripts für die einzelnen Rechner nacheinander auf.
Alle vorgenannten Scripte wurden nur auszugsweise wiedergegeben; die vollständigen Scripte sind im Anhang unter dem Kapitel "Wake-On-Lan Scripte" enthalten.

7.4 Images

Erstellt man ein Abbild (Image) einer Installation eines Rechners, kann man dies mit verschiedenen Programmen erledigen. Diese Programme unterscheiden sich im Wesentlichen in der Art und Weise, wie das Image erstellt und abgespeichert wird. Man unterscheidet grob zwischen dateibasierten Abbildern und Abbildern, die sich an der Geometrie der Festplatte orientieren. Dateibasierte Images sind viel kleiner und schneller zu erstellen als etwa Images

im RAW-Format, die so groß werden können wie die Kapazität der Festplatte, auch wenn sie nur zu einem geringen Teil mit Daten gefüllt ist. Dateibasierte Imageprogramme müssen dafür aber einen Treiber für das Filesystem besitzen, das sie lesen (= Image erstellen) bzw. schreiben (= Image zurückspielen) wollen. Das verwendete Imageprogramm Symantec Ghost benutzt ein dateibasiertes Format und verwendet in den Sicherungsdateien die Dateiendung *.gho*.

Um ein Image zu erstellen, installiert man zuerst den Rechner komplett und startet dann das entsprechende Kloning-Programm. Dies kann sowohl online, offline oder task-gesteuert geschehen. Online bedeutet dabei, dass die Imaging-Software auf dem Rechner selbst installiert wird und in der Lage ist, ein Image im laufenden Betrieb zu erstellen. Als Ziel kann es sich dabei um einen lokalen Datenträger oder ein Netzwerklaufwerk handeln. Manche Imageprogramme sind auch in der Lage, Images im laufenden Betrieb wieder auf den Rechner zurückzuspielen; meist ist dafür aber ein sogenannter separater Task notwendig. Dazu bootet der Rechner neu und startet dabei sein eigenes Betriebssystem und spielt von dort das Image wieder auf den lokalen Rechner zurück. So ist gewährleistet, dass auch die Dateien, die beim normal laufenden Betriebssystem im Zugriff sind, zurückgespielt werden können. Ist der Rechner selbst nicht mehr in der Lage sein Betriebssystem zu booten, dann bleibt noch die Möglichkeit das Image offline auf den Rechner zu bringen. Dazu kann man den Rechner von einem anderen Medium, z.B. CD-ROM oder per PXE-Boot ins Netzwerk, starten und von dort die Imaging-Software starten.

Ein bestimmtes Image ist nur für Rechner mit gleicher oder zumindest ähnlicher Hardware zu verwenden. Will man unterschiedliche Hardware verwenden, muss man verschiedene Images benutzen, je eines für jede Hardwarekategorie. Es ist jedoch mit ein wenig Geschick möglich, ein und dasselbe Image für verschiedene Hardware zu verwenden. Dazu müssen die benötigten Treiber in das Image integriert werden. Dies kann durch Installation ins Betriebssystem geschehen oder durch geschickte Platzierung der Treiber, die dann beim Systemstart dank Plug'n'Play automatisch, d.h. ohne Benutzereingriff, nachinstalliert werden. Nicht alle Möglichkeiten stehen immer zur Wahl. So ist es z.B. notwendig, dass Treiber für den Zugriff auf das Startmedium bereits in dieser Phase des Bootvorgangs zur Verfügung stehen.

7.4.1 Erstellung des Windows XP Images

Die Basis eines Images unter Windows XP ist eine normale Installation des Betriebssystems, in diesem Fall von einer CD, die bereits das Service Pack 2 integriert hat und aus dem Select Rahmenvertrag für den Öffentlichen Dienst bezogen worden ist. Dieser Datenträger ist für Großkunden gedacht und hat den Vorteil, dass Windows nicht aktiviert werden muss. Eine OEM-Version eines Windows XP, wie sie beispielsweise bei vorinstallierten Geräten im Handel erworben werden kann, muss immer einmalig aktiviert werden. Diese Aktivierung kann unter Umständen nach einem größeren Tausch der Hardware nochmals notwendig werden. Wann diese Aktivierung erneut fällig wird, bleibt ein Geheimnis der Firma Microsoft.

Ganz wichtig ist an dieser Stelle, dass weder das Basis- noch das Masterimage jemals einer Domäne hinzugefügt wird. Basisimage bezeichnet hier den Rechner mit der Installation, von dem später das Masterimage erstellt wird, welches auf die zu installierenden Rechner kopiert wird. Dieses Basisimage ist auch die Grundlage, auf der das Image weiterentwickelt wird. Wir werden hier zwei Versionen des Basisimages unterscheiden, zum einen das Ar-

beitsimage und zum anderen das Sysprep-Image oder besser gesagt das Masterimage, das später auf die Rechner verteilt wird. Diese beiden Images unterscheiden sich nur in einer Kleinigkeit, die aber entscheidend für den weiteren Umgang mit dem Image ist.

Um den Überblick zu erleichtern, wird mit Versionsnummern für die Images gearbeitet. Diese Versionsnummern werden hoch gezählt, dabei gibt es keine strenge Vorgabe, wann die Versionsnummer inkrementiert wird. Wurden anfangs die Versionsnummern bei jeder Sicherung des aktuellen Stands hoch gezählt, hat sich in jüngster Zeit ein Versionssprung nur bei gravierenden Softwareerweiterungen oder bei Implementierung einer neuen Rechnerplattform als sinnvoll erwiesen. Den Einstieg macht in dieser Dokumentation die Version 21 und die Entwicklung bis zur Version 22 wird ausführlich dargestellt. Der verwendete Rechnername für das Basisimage lautet *PC60xyz*.

Die grundlegende Hardware, auf der gearbeitet wird und die Entwicklung der Images erfolgt, ist der Rechnertyp FSC Esprimo E 5700, da dieser mit der größten Stückzahl vertreten ist. Die Ausstattung des Rechnertyps ist dabei wie folgt:

- Gehäuse Desktop
- CPU Celeron 420 oder größer
- Arbeitsspeicher 256 MB oder größer
- Festplatte 40GB (IDE oder SATA) oder größer
- Grafikkarte onboard
- DVD-ROM
- 10/100/1000 TX Netzwerkkarte
- 6 USB-Anschlüsse (2x an der Front, 4x hinten)

Als Monitor wird ein TFT mit einer Auflösung von 1.280x1024 Pixeln und 60 Hz Bildwiederholfrequenz verwendet. Die Partitionierung der Festplatte erfolgt dabei mit Laufwerk C: für das Betriebssystem und Standardanwendungen in einer Größe von 20 GB. Der Rest steht als Laufwerk D: zur Reserve für später eventuell lokal benötigte Daten und Programme zur Verfügung. Das Laufwerk E: entspricht dem DVD-ROM Laufwerk.

Auf diesen Rechner wird das Betriebssystem samt Service Pack[21] 2 und benötigten Hotfixes[22] installiert. Anschließend werden die notwendigen Treiber für die Hardware eingespielt, wie z.B. Grafikkarte, Netzwerkkarte, Sound, USB, Massenspeicher. Für den Zugriff auf Peripheriegeräte, wie Drucker und Scanner, werden ebenfalls Treiber und Bediensoftware eingespielt.

Die Konfiguration von betriebssystemnahen Diensten, wie "Firewall" oder "Automatische Updates"[23], sollte nun erfolgen, um spätere Probleme mit Anwendungen oder ungewünschte automatische Updates zu verhindern. Es ist sinnvoll an dieser Stelle beide Dienste nicht automatisch zu starten, sondern die Default-Einstellung von "automatisch" auf "manuell" zu ändern. Eine detaillierte Konfiguration dieser Dienste kann zu einem späteren Zeitpunkt, wenn das Gerät Mitglied einer Domäne ist, von einer zentralen Stelle im Active Directory über Richtlinien erfolgen. Im Übrigen empfiehlt es sich generell, so wenig wie möglich an den Default-Einstellungen zu ändern. Alles, was zentral über Richtlinien eingestellt werden kann, sollte auch darüber abgewickelt werden.

---

[21] Eine Sammlung von Hotfixes oder Funktionserweiterungen, die von Microsoft gelegentlich als Gesamtwerk zur Verfügung gestellt wird.

[22] Fehlerbehebungen oder Sicherheitsupdates, die gezielt einzelne Fehler oder Sicherheitslücken beheben.

[23] Ein Dienst, der automatisch Hotfixes und Updates herunterladen und einspielen kann.

Nicht benötigte betriebssystemeigene Software, wie Outlook Express oder Windows Messenger, beides sind Light-Varianten von Mail-Clients, sollten deinstalliert werden, um Probleme mit dem Standard Mail-Client Outlook 2003 zu verhindern.
Anschließend kann damit begonnen werden, sonstige Software auf den Rechner zu installieren, die nachher auf allen Endgeräten verfügbar sein soll und als sogenannte Standardanwendungen bezeichnet werden. Standardanwendungen sind:

- Microsoft Office 2003 Professional (Officepaket)
- Outlook 2003 aus dem Paket Microsoft Office (Mail-Client)
- Internet Explorer 6 (Browser)
- Sun Java Runtime Environment 4 (Browsererweiterung und Java-Plattform)
- Adobe Acrobat Reader 7 (PDF-Betrachter)
- FreePDFXP 3 (PDF-Drucker)
- Irfanview 3 (Bildbetrachter)
- Pack- und Entpackfunktionalität über die betriebssystemeigene Komponente (zip)
- Citrix ICA-Client 9 (Zugriff auf Terminalserver)
- Snapshot (zur komfortablen Erstellung von Printscreens)

Als Hintergrundbild wird ein CD-konformes[24] und selbst erstelltes Bild in jpg-Format verwendet. Dieses Hintergrundbild ist in einer Größe von 1.024x768 Bildpunkten angelegt und so entworfen, dass es auch in einer gestreckten Darstellung von 1.280x1.024 noch angenehm für die Augen ist. Weitere für den Support wichtige Programme werden eingespielt, wie NSM Netsupport Manager[25] für den Fernzugriff sowie BGInfo[26] zur unterstützenden Darstellung von Supportinformationen. Diese Informationen können von dem Endanwender mittels einer Verknüpfung über "Start/wer-und-wo-bin-ich" abgerufen werden und werden als Popup-Fenster realisiert.

[24] CD = Corporate Design, bei der KVMYK existiert ein eigenes Handbuch mit konformen Regeln zur Gestaltung von Elementen.

[25] Kostenpflichtige Software, *http://www.netsupportmanager.com/DE/index.asp*

[26] Kostenfrei von Microsoft, *http://www.microsoft.com/technet/sysinternals/Utilities/BgInfo.mspx*

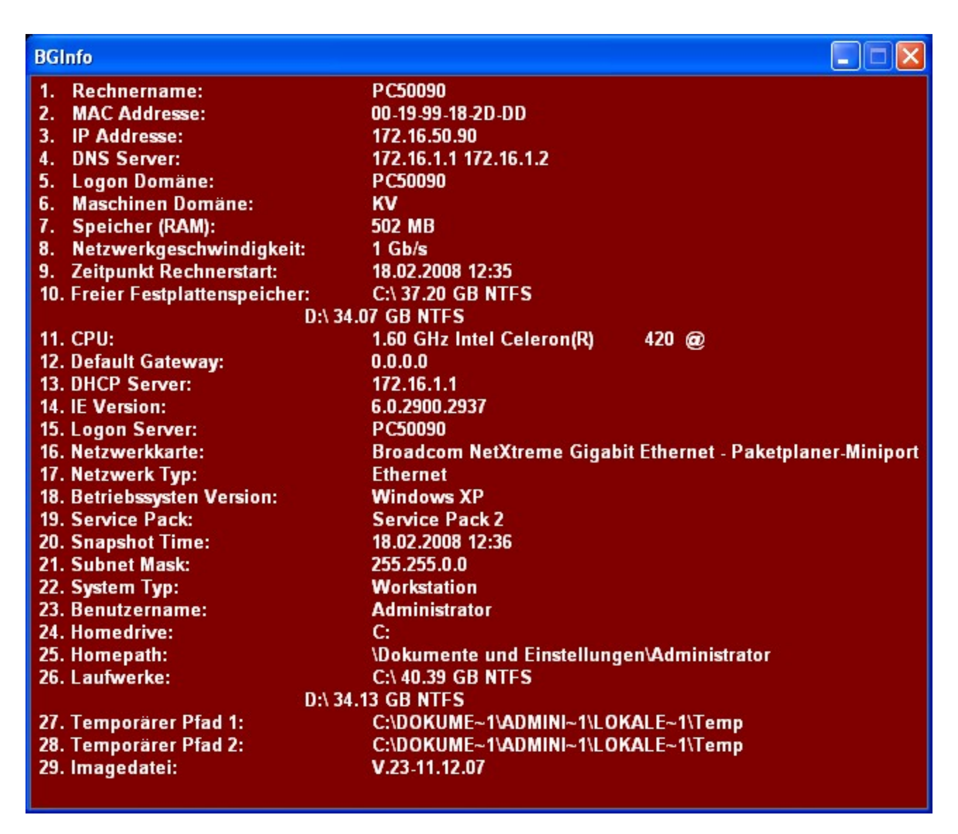


Abbildung 7.12 Supportinformation "wer-und-wo-bin-ich"

Zur Dokumentation und lokalen Verfügbarkeit der verwendeten Treiber wird auf Laufwerk C: ein Verzeichnis namens *Install* angelegt, in welches die Treiber abgelegt werden. Dies wird auch später von Nutzen sein, wenn dort vor der Transformation des Images auf eine neue Hardwareplattform bereits die notwendigen neuen Treiber hineinkopiert werden. Damit sind ein Brennen einer CD und das Aufspielen der Treiber über ein CD-Laufwerk nicht erforderlich. Im Weiteren existieren unterhalb des Verzeichnisses *Install* noch die eingespielten Hotfixes im Verzeichnis *Hotfix_XP*, die Druckertreiber im Verzeichnis *Printer*, das Handbuch des E 5700 im Verzeichnis *E5700_Handbuch* sowie die Monitorinformationsdatei und das Handbuch für den Monitor FSC Scenicview P19-2 im Verzeichnis *Monitor*.

Nun werden noch diverse Systemeinstellungen vorgenommen, so zum Beispiel die Einstellungen der Auslagerungsdatei und sonstige relevante Konfigurationen. Die Auslagerungsdatei entspricht der Swap-Partition bei Linux und dient zum Auslagern von (zurzeit nicht benötigten) Seiten des Arbeitsspeichers und wird somit zur Vergrößerung des RAM-Bereichs verwendet. Weiterhin werden diverse Bereinigungen und Performanz steigernde Maßnahmen durchgeführt. Dazu zählt das Bereinigen des DLL-Caches, den Microsoft im Rahmen der "Automatischen Systemwiederherstellung" anlegt. Die "Automatische Systemwiederherstellung" überwacht das Austauschen von betriebssystemeigenen DLLs, welches durch Installation von Drittanbietersoftware immer wieder passiert, und legt zudem einen Wiederherstellungspunkt an. Sollte also eine Installation fehlschlagen oder zu Problemen führen, kann über diese Wiederherstellung der Zustand vor der Installation wieder herbeigeführt werden. Da die von uns verwendete Software vorab umfangreich getestet worden ist, kann man diesen Cache und den Wiederherstellungspunkt nach neuen Installationen löschen. Das Bereinigen des Caches geschieht über den Befehl *sfc /purgecache* und das Löschen der Wiederherstellungspunkte wird im Rahmen der Datenträgerbereinigung durchgeführt. Die Datenträgerbereinigung, die unter dem Menüpunkt Systemverwaltung im Zubehör zu

finden ist, löscht nicht mehr benötigte Dateien von Laufwerken und bietet die Möglichkeit alle Wiederherstellungspunkte, bis auf den letzten, zu entfernen. Nach dem Start des Programms und dem Auswählen des Ziellaufwerkes C: wird eine Berechnung durchgeführt und anschließend erscheint eine Übersicht.

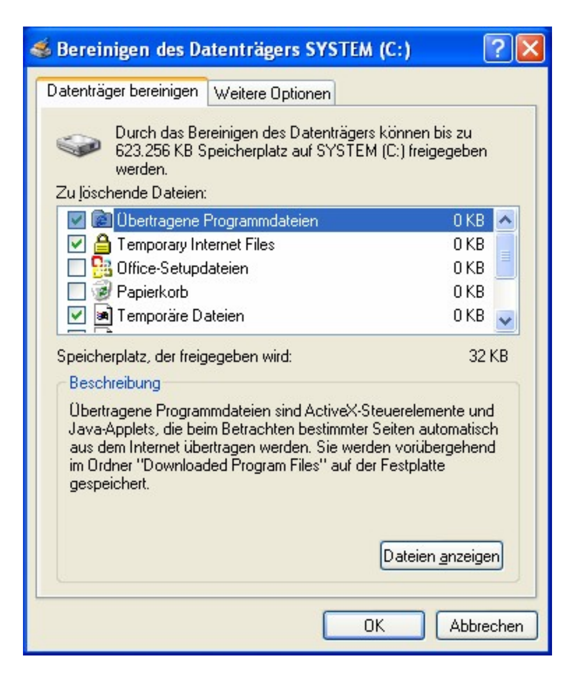


Abbildung 7.13 Datenträgerbereinigung

Unter dem Punkt "Weitere Optionen" kann nach einer Sicherheitsabfrage die Bereinigung der Systemwiederherstellung durchgeführt werden.

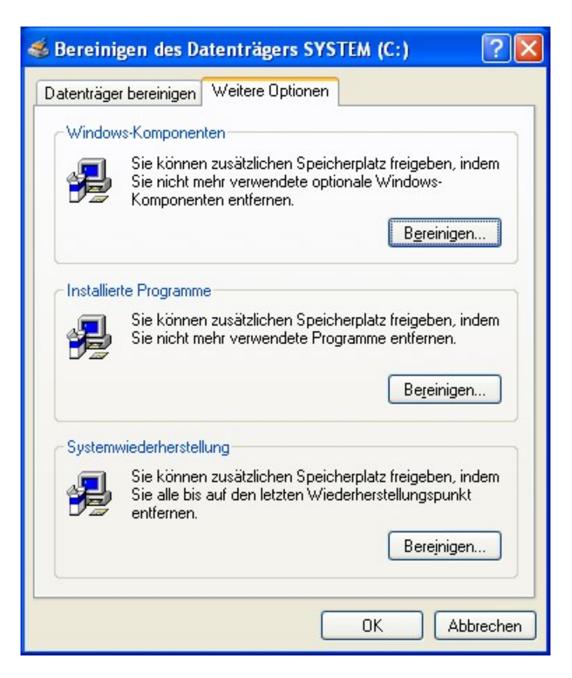


Abbildung 7.14 Systemwiederherstellung bereinigen

Anschließend sollte die Festplatte, insbesondere Laufwerk C:, defragmentiert werden. Dies geschieht grafisch über die Computerverwaltung oder den Aufruf von *defrag c:* in der Konsole. Eine geringe Fragmentierung sorgt für schnelleres Laden von Programmen und Daten, da

Zugriffe optimiert werden. Unter Defragmentierung versteht man die Neuordnung von Datenblöcken fragmentierter Dateien auf den Spuren und Sektoren der Festplatte. Die vorgenannten Optionen zur Bereinigung und Steigerung der Performanz werden regelmäßig vor jeder Erstellung einer neuen Version eines Images durchgeführt. Dies führt außerdem zu einer Verringerung der Dateigröße des Images.
Damit ist das Image inhaltlich fertig. Was nun folgt, ist Vorbereitung für die Ausbringung und die Automatische Installation des Images auf der Zielhardware. Um die Übersichtlichkeit zu verbessern, ist der Prozess in die folgenden 4 Teilschritte untergliedert worden:

I. Vorbereitung mittels *Sysprep*
II. Individualisierung mittels *ip2name.cmd*
III. Erweiterung des *Sysprep*-Prozesses
IV. Zusätzliche Automatisierung

I. Vorbereitung mittels *Sysprep*

Microsoft liefert auf der Installations-CD im Ordner *\Support\Tools* in der gepackten Datei *deploy.cab* ein Tool namens *sysprep.exe* mit. Dieses Tool dient dazu, die Verteilung von Windows XP zu automatisieren. Dazu werden nach einem Aufruf von *sysprep*

- alle individuellen Einstellungen der Installation gelöscht,
- eine neue SID generiert,
- der Rechnername geändert (mit Einschränkungen),
- der Rechnerbeitritt zu einer Domäne oder Workgroup automatisiert,
- der Rechner neu gestartet,
- eine automatische Anmeldung durchgeführt.

Der grundlegende Ablauf sieht dabei folgendermaßen aus:

1. Arbeitsimage *PC60xyz22.gho* erstellen
2. Ordner *\Sysprep* erstellen und alle *Sysprep*-Dateien hinein kopieren
3. *sysprep.exe* aufrufen
4. Rechner herunterfahren
5. Masterimage *PC60xyz22_sysprep.gho* erzeugen

Dabei ist das Image *PC60xyzVersionsnummer.gho* das Arbeitsimage und *PC60xyzVersionsnummer_sysprep.gho* das zugehörige zu verteilende Masterimage. Dann wird dieses Image auf die Zielrechner verteilt. Die Grafik veranschaulicht den Ablauf.

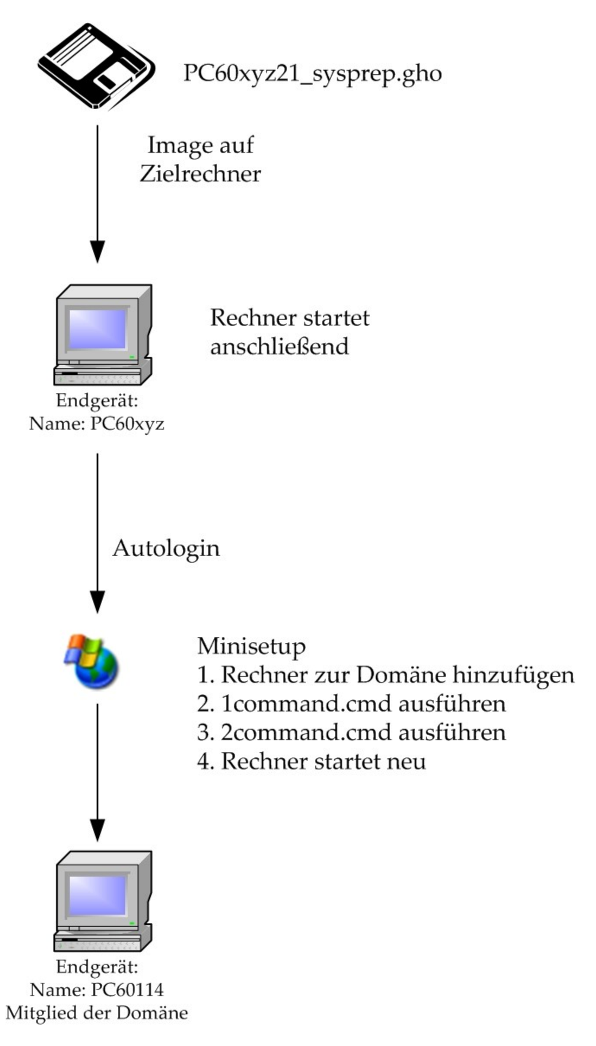


Abbildung 7.15 normaler *Sysprep*-Vorgang

Zu *sysprep.exe* gehören zwingend die Dateien *setupcl.exe* und die *sysprep.inf*. Die Datei *setpcl.exe* ist ebenfalls Teil der *deploy.cab* und die *sysprep.inf* kann über einen Assistenten mit dem *setupmgr.exe*, der ebenfalls in der *deploy.cab* enthalten ist, erstellt werden. Die Datei *sysprep.inf* kann aber auch mit einem beliebigen Texteditor erstellt werden. Ist die Datei *sysprep.inf* nicht vorhanden, kann keine vollautomatische Installation erfolgen, da dann bei jeder Installation eines Rechners die fehlenden Informationen über die Tastatur eingegeben werden müssen. Hinweise und Hilfe zum Umgang mit *sysprep* finden sich in den beiden Windows Hilfedateien *ref.chm* und *deploy.chm* in der gleichnamigen Datei mit der Endung cab. *Sysprep.exe* kann mit diversen Parametern aufgerufen werden, die in der Folge erläutert werden, und muss zwingend mit der *setupcl.exe* und *sysprep.inf* im Verzeichnis *C:\Sysprep* liegen.

Dabei enthält die Datei *sysprep.inf* Steuerinformationen über den Vorbereitungsprozess, wie z.B. die Lizenzinformationen der verwendeten Installation und muss für eine vollautomatische Installation mindestens die folgenden Abschnitte und Eintragungen enthalten: [MSR04]

Abschnitt [GuiUnattended]

AdminPassword=*	Setzt das Passwort für den Administrator; *= leeres Kennwort
OEMSkipRegional=1	Regions- und Sprachoptionen während der Installation anzeigen; 1=aus; 0=ein
TimeZone=110	Zeitzone setzen; 110= West-Europa/Berlin

Abschnitt [BenutzerData]

FullName="KVMYK"	Name des Benutzers
OrgName="Kreisverwaltung Mayen-Koblenz"	Name der Firma
ComputerName=Computer1	Name des Computers; ist der Eintrag leer, wird er während der Installation abgefragt; lautet der Eintrag "*", vergibt Windows während der Installation einen zufälligen Namen

Abschnitt [Identification]

JoinDomain=Domaene	Name der beizutretenden Domäne
DomainAdmin=administrator	Name des administrativen Accounts der Domäne
DomainAdminPassword=administrator	Kennwort des administrativen Accounts der Domäne

Der Vorbereitungsprozess wird durch *sysprep.exe –mini –activated –reseal –quiet* gestartet und der Rechner fährt anschließend automatisch herunter.
Dabei bedeuten die Optionen: [MSD04]

mini	Konfiguriert Windows XP Professional so, dass statt der Windows-Willkommensseite die Miniinstallation verwendet wird.
activated	Die Kulanzfrist für die Windows-Produktaktivierung wird nicht zurückgesetzt und der Rechner bleibt aktiviert.

reseal	Löscht die Protokolle der Ereignisanzeige und bereitet den Computer für die Auslieferung an den Kunden vor. Beim nächsten Start wird die Windows-Willkommensseite bzw. die Miniinstallation gestartet.
quiet	Führt Sysprep ohne Anzeige von Bestätigungsmeldungen auf dem Bildschirm aus.

Der "Haken", der gegen eine vollautomatische Verteilung auf viele Rechner spricht, ist offensichtlich. Es muss ein Computername definiert werden. In der vorgenannten *sysprep.inf* erhalten alle Installationen den Rechnernamen *Computer1*. Dies ist aber in einem Netzwerk nicht zulässig, dort müssen Rechnernamen eindeutig sein. Dies bedeutet, dass nur die Möglichkeit besteht, dort einen * einzutragen und somit den Computernamen dem Zufallsgenerator zu überlassen. Den Computernamen leer zu lassen bedeutet den Verlust der Vollautomatisierung und verlangt die manuelle Eingabe im Laufe der Installation.

II. Individualisierung mittels *ip2name.cmd*

Will man Einfluss auf den Namen haben, muss der Computername dynamisch zur Laufzeit der Installation und vor dem Aufruf der *sysprep.exe* in die Datei *sysprep.inf* eingetragen werden.

Dazu wird der zuvor dargestellte grundlegende Ablauf geändert.

1. Arbeitsimage *PC60xyz22.gho* erstellen
2. Ordner *Sysprep* erstellen und alle *Sysprep*-Dateien hinein kopieren
3. Autologin eintragen
4. Rechner herunterfahren
5. Masterimage *PC60xyz22_sysprep.gho* erzeugen

Der Kern der Idee besteht darin, dass nicht der *Sysprep*-Prozess gestartet wird, sondern man setzt eine Stufe früher an. Wird also das Image über PXE an den Zielrechner verteilt, startet dieser mit Windows XP und meldet sich automatisch[27] mit einem Benutzer mit administrativen Rechten an der Station an. Nach dieser Anmeldung wird direkt ein Script gestartet, welches als erstes den Computernamen in die *sysprep.inf* einträgt und anschließend den *Sysprep*-Prozess startet. Die Individualisierung, also das Eintragen eines eindeutigen Computernamens in die *sysprep.inf*, erfolgt erst zu dem Zeitpunkt, wo das Image bereits auf der Zielhardware ist.

Zu dem Zeitpunkt, wo das automatische Anmelden erfolgt, besitzt der Rechner Netzverbindung und hat durch eine Reservierung (vgl. Kapitel 7.2 DHCP-Server allgemein) eine eindeutige IP-Adresse zugewiesen bekommen. Aufgrund der Namenskonventionen (vgl. Kapitel 6.1.2 Namenskonventionen und Versionierung) und der IP-Adresse kann der Rechnername eindeutig aus diesen Informationen erstellt werden. Lautet eine IP-Adresse 172.16.60.114, wird daraus automatisch der Rechnername *PC60114* ermittelt. Die Arbeitsweise dieses Scripting-Vorgangs wird genauer in Kapitel 7.7 und 7.8 dargestellt.

---

[27] Über das Setzen von Registrykeys oder mit "Autologon for Windows" von Sysinternals *http://www.microsoft.com/technet/sysinternals/Security/Autologon.mspx*

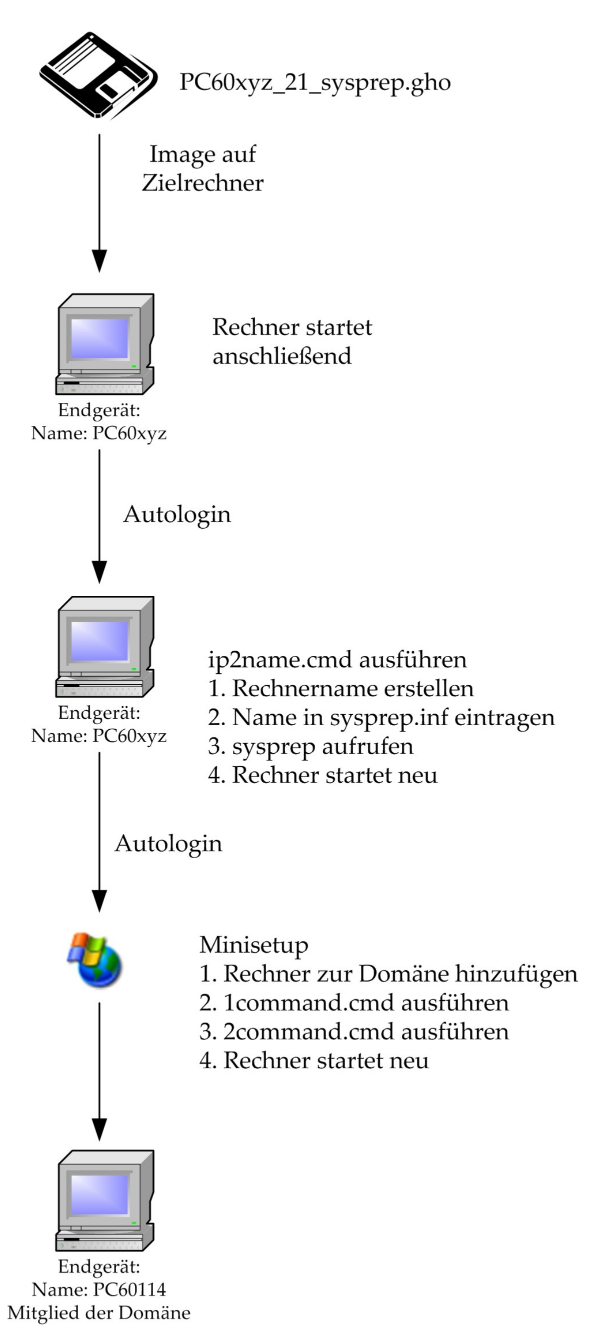


Abbildung 7.16 erweiterter *Sysprep*-Vorgang

III. Erweiterung des *Sysprep*-Prozesses

Die Datei *sysprep.inf* wird noch um folgende Einträge erweitert:

Abschnitt [unattended]

OemSkipEula=Yes	Abwicklung der EULA (= End-Benutzer License Agreement} ; yes = übergehen; no = nicht übergehen
InstallFilesPath=C:\sysprep\i386	Pfad für die Dateien während des Minisetups.

Abschnitt [GuiUnattended]

OEMDuplicatorstring=PC60xyz23-Basisimage-Sysprep-111207	Wird als Zeichenkette in der Registry unter HKLM\System\Setup\OemDuplicatorString eingetragen und hat nur informativen Charakter. Anmerkung: HKLM= HKEY_LOCAL_MACHINE
OemSkipWelcome=1	Willkommensseite im mini-setup anzeigen; 1 = nein; 0 = ja

Abschnitt [BenutzerData]

ProductKey=AAAAA-12345-BBBBB-67890-CCCCC	Produktschlüssel für die Installation

Abschnitt [Networking]

InstallDefaultComponents=Yes	Installation der Default-Komponenten; Yes = ja, No = nein Anmerkung: Der Abschnitt muss vorhanden sein, der Eintrag InstallDefaultComponents kann weggelassen werden, da der default=yes ist. [MSR04]

Besonders wichtig ist an dieser Stelle der Eintrag InstallFilesPath=*C:\sysprep\i386*, da dort im Unterverzeichnis *OEM* nach der Datei *cmdlines.txt* gesucht wird. In diese Datei kann man alles eintragen, was im Anschluss an das Minisetup aufgerufen werden soll. Sie enthält die beiden folgenden Einträge:

```
[Commands]
"C:\Sysprep\1command.cmd"
"C:\Sysprep\2command.cmd"
```

Die beiden Scriptdateien *1command.cmd* und *2command.cmd* haben folgenden Inhalt:

```
@ECHO OFF
REM 1command.cmd
REM HT/07.07.2006   :Cleanup
REM Sysprep - unnoetige Treiber bereinigen
ECHO 1. Sysprep aufraeumen.
@C:\Sysprep\Sysprep –clean

REM 2command.cmd
REM HT/07.07.2006
ECHO.
```

In der *1command.cmd* wird *sysprep.exe* mit dem Parameter "-clean" aufgerufen. Der Parameter "-clean" sorgt dafür, dass alle nicht mehr benötigten Massenspeichertreiber aus der Registry gelöscht werden. Dies beschleunigt den Bootvorgang und damit den Systemstart des Rechners. In der *2command.cmd* wird derzeit nichts ausgeführt.

IV. Zusätzliche Automatisierung

Es gibt darüber hinaus bei dem *Sysprep*-Vorgang die Möglichkeit, zusätzliche Tasks automatisch durchführen zu lassen. Dazu zählt auch das automatische Anmelden, nachdem der eigentliche *Sysprep*-Vorgang bereits erledigt ist, der Zielrechner bereits Mitglied der Domäne ist und Benutzer sich anmelden könnten. Dieser administrative Login bietet die Möglichkeit zum Abschluss noch Tätigkeiten auszuführen, die erst nach dem Beitritt zu einer Domäne möglich sind. Dazu zählt beispielsweise das Aufnehmen von Domänenbenutzern in lokale Gruppen des Rechners. Bereits beim Zutritt zur Domäne wird die Gruppe der Domänenadministratoren Mitglied der Gruppe der lokalen Administratoren auf jedem PC. Dieser Vorgang geschieht immer, wenn ein PC Mitglied in einer Domäne wird. Genauso wird diese Gruppe beim Verlassen der Domäne wieder entfernt.

Man nutzt die Möglichkeit über den einmaligen Login (AutoLoginCount=1), um einen Domänenbenutzer zum Mitglied der lokalen Administratoren eines PCs zu machen. Dieser Benutzer kann damit auf dem PC einen Dienst starten, der ggf. lokale Software installieren kann und zeitgleich Zugriff aus das Netz bzw. Dateifreigaben hat. Dieser Domänenbenutzer heißt in unserem Fall *PCinst* und soll einen gleichlautenden Dienst auf dem PC bedienen, der beim Systemstart automatisch startet. Dieser Dienst startet im Verzeichnis *C:\Install\PCinst* das Script *install.cmd*, welches alle Scripts (mit der Endung *.cmd*) im Unterverzeichnis *InstallerScripts* startet. So ist es möglich, ohne kostenpflichtiges Zusatzprogramm Softwareverteilung in geringem Umfang zu realisieren. Im Rahmen des Active Directory und unter Ausnutzung von Richtlinien ist es zwar möglich, eine simple Softwareverteilung zu realisieren, die allerdings auf Basis des Microsoft eigenen Windows Installers arbeitet und auf Installationen beschränkt ist, die im Format eines MSI-Paketes (MSI = Microsoft Installer) vorliegen. Dies ist leider nicht für alle Softwareinstallationen der Fall.

Den automatischen Login erreicht man durch den Eintrag "AutoLogon" in der Datei *sysprep.inf*.

Abschnitt [GuiUnattended]

AdminPassword=administrator	Passwort des lokalen Administrator-Accounts
EncryptedAdminPassword=No	Verschlüsselung für das Administratorpasswort; yes = verschlüsselt; no= nicht verschlüsselt; Anmerkung: Ist das Passwort verschlüsselt, ist kein Autologin mehr möglich.
AutoLogon=Yes	Konfiguriert die automatische Anmeldung mit dem Administrator; Yes = aktiviert, No = deaktiviert
AutoLogonCount=1	Anzahl der automatischen Anmeldungen

Abschnitt [GuiRunOnce]

Command0=C:\Install\Sysprep\1login.cmd	Beim ersten Anmelden diese Befehle ausführen.

Die oben dargestellten Optionen bewirken das einmalige automatische Anmelden mit dem Benutzer *Administrator*. Dieser wird allerdings mit dem in der *sysprep.inf* genannten Kenn-

wort (AdminPasswort=) angemeldet, allerdings muss das Kennwort des lokalen Administrators zum Zeitpunkt der automatischen Anmeldung über den sysprep-Prozess leer sein, d.h. es darf keines vergeben sein. Das in der Option AdminPassword=*administrator* eingestellte Kennwort wird erst nach Abschluss des gesamten Vorgangs eingestellt. Kapitel 7.7 geht nochmals im Zusammenhang mit dem Batch-Script *ip2name.cmd*. detaillierter auf das Thema Kennwörter ein, das zugegebenermaßen an dieser Stelle recht verwirrend klingt.
Nach dem automatischen Login wird das Script *C:\Install\Sysprep\1login.cmd* aufgerufen. Wichtig ist hierbei, dass der aufzurufende Befehl sich nicht im Verzeichnis *C:\Sysprep* befinden darf, da das zu diesem Zeitpunkt bereits automatisch durch den *Sysprep*-Vorgang gelöscht worden ist. Der Inhalt des Scriptes *1login.cmd* ist hier dargestellt, wobei Befehle optischer oder informativer Natur der Übersichtlichkeit weggelassen worden sind. Das komplette und ausführliche Script findet sich im Anhang wieder.

```
@ECHO OFF
(1)@start /wait net localgroup administratoren KV\PCinst /add
(2)@start /wait reg import %InstDir%\Sysprep\autologon\
autologin_ohne_username.reg >nul
(3)@start /wait %InstDir%\Sysprep\instservice.cmd
(4)@start /wait %InstDir%\PCinst\pcinst.cmd
(5)@start /wait %InstDir%\Sysprep\dsmodpc.cmd
(6)@start /wait %InstDir%\session\regtlb \\DBSRV8\Session$\SmcServer.tlb
(7)@start %InstDir%\Sysprep\shutdown.exe /L /R /T:10 "Reboot laeuft" /C
(8)start C:\install\Sysprep\2.cmd
```

Zusammenfassend lässt sich sagen, dass in dem Script,

- (1) der Benutzer *KV\PCinst* Mitglied der lokalen Administratoren wird,
- (2) Registrierungseinstellungen importiert werden, die für den nächsten Start den letzten Anmeldenamen ausblenden und den Autologin deaktivieren,
- (3) der Dienst *pcinst* installiert wird,
- (4) das Script *pcinst.cmd* gestartet wird,
- (5) der PC Mitglied einer bestimmten Gruppe wird,
- (6) die Arbeitsstation für den Einsatz einer bestimmten Fachsoftware vorbereitet und eine Änderung der Registry durchgeführt wird,
- (7) ein Neustart mit einem Countdown von 10 Sekunden gestartet wird,
- (8) das Script *2.cmd* gestartet wird, das alle nicht benötigten und sicherheitskritischen Scripte, die z.B. Kennwörter enthalten, löscht.

Das Script *pcinst.cmd* macht nichts anderes, wie später auch der entsprechende Dienst *PCinst*. Zu diesem Zeitpunkt kann der Dienst nicht gestartet werden, da noch nicht alle Richtlinien verarbeitet worden sind. Somit besitzt der Benutzer *PCInst*, der der zugehörige Sicherheitsaccount des Dienstes *PCinst* ist, noch nicht die erforderlichen Berechtigungen zum Start eines Dienstes. Daher bringt man in diesem Fall einmalig die *pcinst.cmd* zur Ausführung. Die Ausführung des Scriptes *2.cmd* ist eine reine Sicherheitsmaßnahme, da in den Scripten, die von der *1login.cmd* aufgerufen werden, die Kennwörter der Benutzerkonten des lokalen *Administrators* sowie des lokalen Benutzers *PCinst* enthalten sind. Danach startet der Rechner nochmal und ist für den Endanwender einsatzbereit. Das Script *dsmodpc.cmd* nimmt den PC in eine bestimmte Gruppe im Active Directory auf, die dafür sorgt, dass die notwendigen Richtlinien und Berechtigungen für den PC erteilt werden, damit unter anderem der Dienst *PCinst* gestartet werden kann.

Kapitel 7.7 wird ausführlich auf die Automatisierung zu sprechen kommen und den kompletten Ablauf visuell als auch tabellarisch aufbereitet darstellen.

7.4.2 Erstellung des eLux NG Images

Die Erstellung oder besser gesagt die Definition eines Images mittels einer *idf*-Datei geschieht durch ELIAS NG, das entweder separat oder in der Verwaltungskonsole Scout NG an entsprechender Stelle[28] aufgerufen werden kann. [ELI04]
Nach dem Start erscheint folgende Übersicht über die gewählten und zur Verfügung stehenden Softwarepakete.

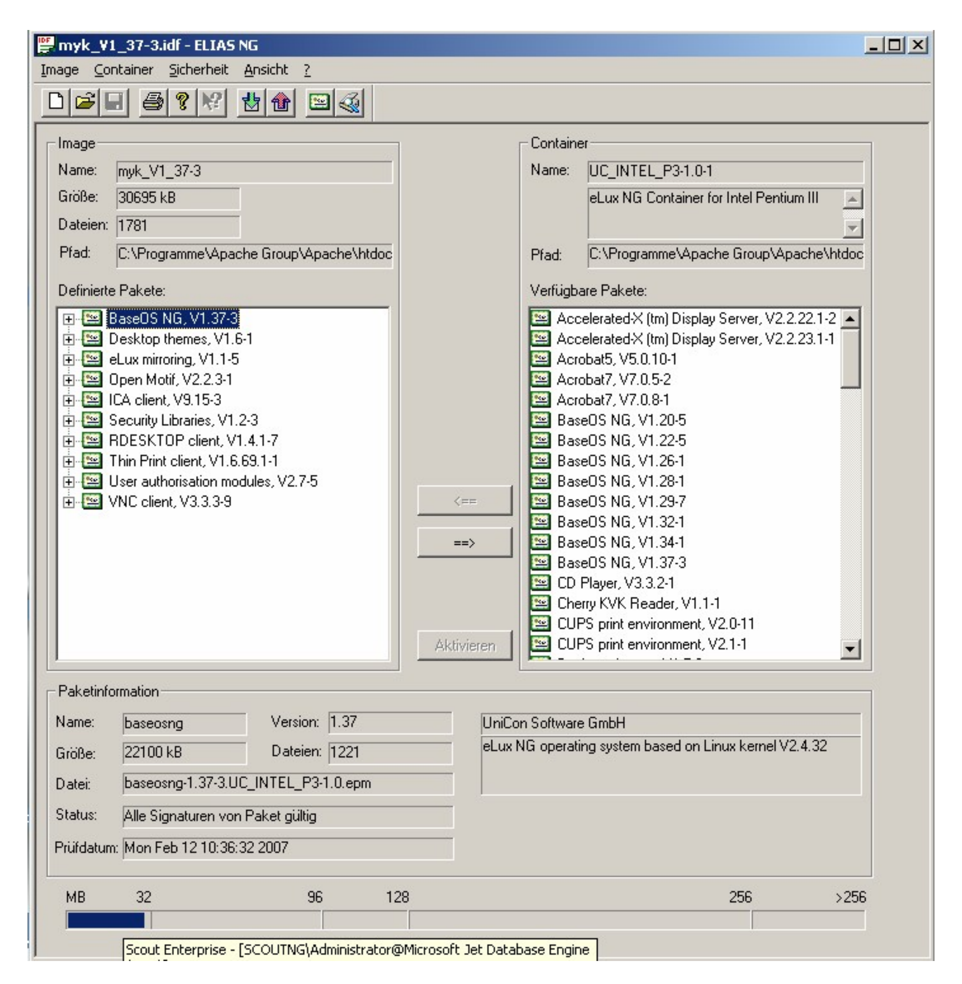


Abbildung 7.17 ELIAS NG

Die hier abgebildeten Pakete sind im Wesentlichen das Basisbetriebssystempaket "BaseOS NG, V1.37-3" und der ICA-Client "ICA client, V9.15.3", der das Paket "Open Motif, V2.2.3-1" benötigt. Die anderen Pakete sind für Zugriff auf Citrix-Terminalserver nicht unbedingt notwendig, dienen eher optionalen Komponenten wie z.B. Zugriff auf reine Windows Terminalserver über das Remote-Desktop-Protocol "RDESKTOP client, V1.4.1-7" oder zur Zeit nicht eingesetzten Features, wie das Authentifizieren gegen das Active Directory "Benutzer authorization modules, V2.7-5". Wichtig ist bei der Definition eines Images, dass man sich zuerst auf einen Container festlegt, wobei hier UC_INTEL_P3 und UC_PC zur Auswahl stehen. Der Container bestimmt die primäre Plattform, also zertifizierte Intel-Hardware (UC_INTEL_P3) oder den generellen PC-Container UC_PC für alle PCs, die nicht mit dem UC_INTEL_P3 Container funktionieren oder davon unterstützt werden. Das Programm lie-

[28] Unter dem Punkt Konfiguration/Firmware einer Organisationseinheit oder entsprechender Stelle in der Basiskonfiguration.

fert zudem unterhalb der Definition des Images noch Informationen über die aktuelle Größe und die Anzahl der Dateien im gesamten Image. Die hier gezeigte Konfiguration "myk_V1_37-3.idf" hat eine Größe von ca. 30 MB. Die Datei *myk_V1_37-3.idf* muss in dem zugehörigen Containerverzeichnis, hier *C:\Programme\Apache Group\Apache\htdocs \eluxng\UC_INTEL_P3*, abgespeichert werden.

Damit diese Konfiguration von den zu installierenden Thin-Clients benutzt wird, muss sie an geeigneter Stelle zugewiesen werden. Dies geschieht in der Verwaltungskonsole Scout NG im Menüpunkt der "Recovery-Einstellungen" unterhalb des Menüpunktes "Optionen".

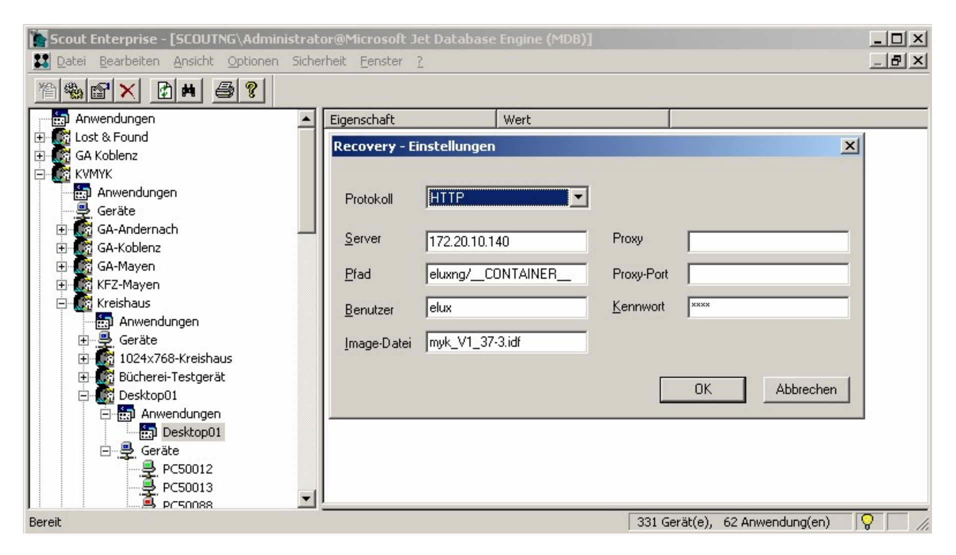


Abbildung 7.18 Recovery-Einstellungen im Scout NG

Die hier eingestellten Optionen werden dann anschließend in die in Kapitel 7.3.2 genannte Datei *eluxng.des* übertragen und beim Booten des Thin-Clients ausgelesen.

Die Verwaltungskonsole Scout NG

Sie dient der strukturierten Ablage der Clientgeräte und bietet die Möglichkeit, geschachtelte Organisationseinheiten zu bilden, in denen Geräte und Anwendungen gruppiert werden. So werden Geräte mit gleicher Konfiguration und gleichen Anwendungen unter Ausnutzung der Vererbungshierarchie sinnvoll gruppiert, ähnlich dem des Active Directory. Dabei bildet die Basiskonfiguration die oberste Ebene.

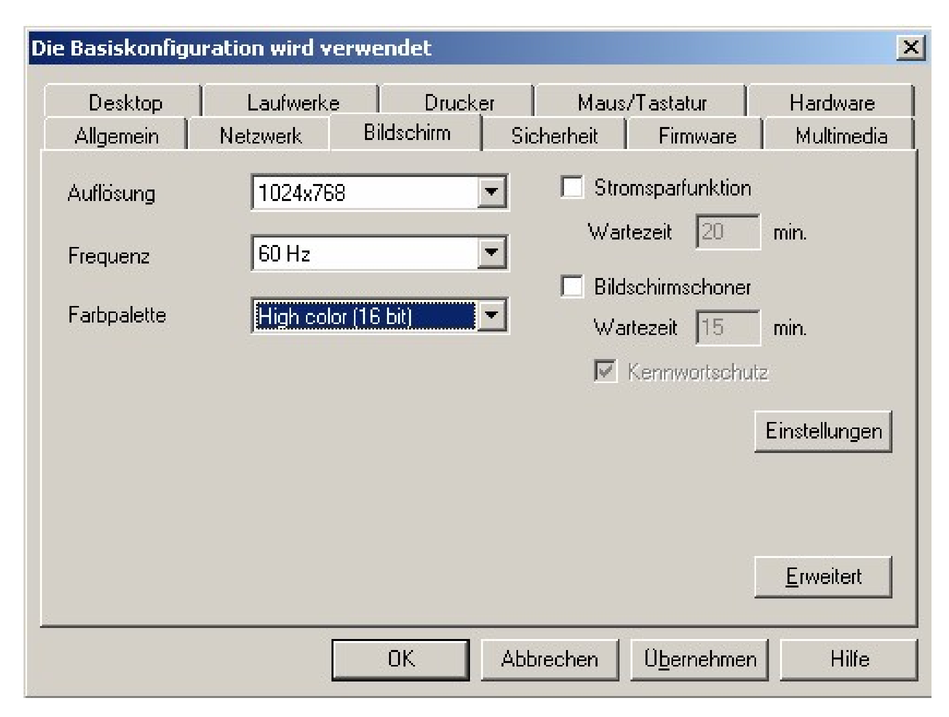


Abbildung 7.19 Basiskonfiguration in der Verwaltungskonsole Scout NG

Hier werden also die Eigenschaften des Gerätes und damit die Einstellungen der Software des Gerätes vorgenommen, z.B. die Auflösung, Farbtiefe und die Bildwiederholfrequenz. Die Basiskonfiguration wird dabei auf die in Bild 7.20 sichtbaren OUs "Lost & Found", "GA Koblenz" und "KVMYK" und deren untergeordnete OUs vererbt. Es ist jederzeit möglich, die Vererbung auf Ebene einer OU oder sogar eines einzelnen Gerätes zu durchbrechen und individuelle Einstellungen vorzunehmen. Ebenso werden auf oberster Ebene definierte Anwendungen in der Hierarchie absteigend weiter vererbt, wenn dies gewünscht wird.
Jede OU erhält eine eindeutige ID, in unserem Beispiel die 28.

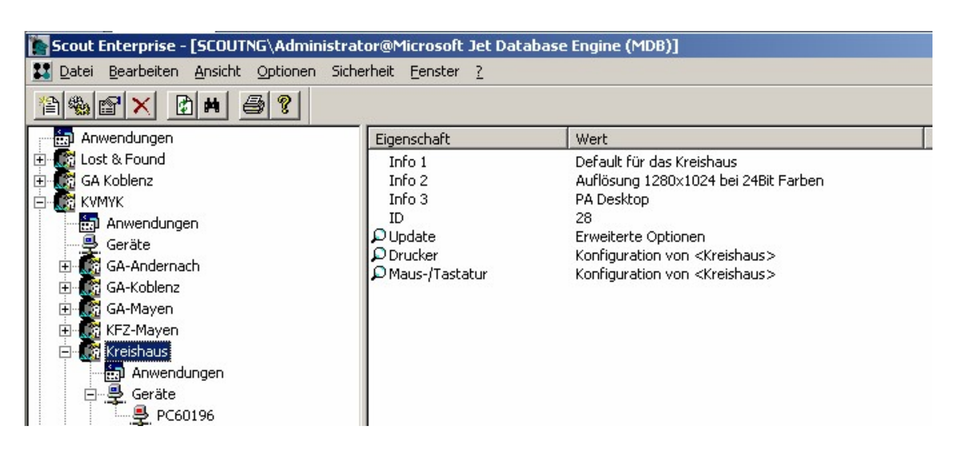


Abbildung 7.20 Organisationseinheiten in der Verwaltungskonsole Scout NG

Diese ID wird in den DHCP-Optionen mit Nr. 223 hinterlegt und von den Thin-Clients bei deren Installation ausgelesen. Der Client, in unserem Beispiel *PC60196*, wird automatisch dieser Organisationseinheit zugeordnet und taucht dort nach Fertigstellung der Installation auf. Da die Vererbung der Konfiguration und der Eigenschaften per Default für das Gerät eingeschaltet ist, erhält der Client automatisch die entsprechende Konfiguration und die Anwendungen der OU zugeordnet. Der Thin-Client ist damit direkt einsatzbereit und fertig.

7.4.3 Erstellung eines Server Images

Die grundsätzliche Vorgehensweise ist hier analog zum Kapitel "Erstellung des Windows XP Images". Der Unterschied liegt einerseits in der verwendeten Hardware und natürlich in der Betriebssystemversion. Die Ausstattung der Terminalserver HP Proliant DL360 G4 ist dabei wie folgt:

- Gehäuse 19" Rack
- 2x CPU Intel XEON 3,0 GHZ
- Arbeitsspeicher 4 GB
- Festplatte 2x 36GB (SCSI mit RAID-Level 1)
- Grafikkarte onboard
- DVD-ROM
- Dual 10/100/1000 TX Netzwerkkarte
- 4 USB-Anschlüsse (2x an der Front, 2x hinten)

Als Betriebssystem wird Windows Server 2003 R2 Standard deutsch verwendet und die Installation erfolgt dabei über einen Select Datenträger. Die Installation läuft hier nicht manuell, sondern geschieht über ein Utility, das dem HP Server beigefügt ist. Dieses nennt sich "HP Smartstart". Diese CD wird beim ersten Start des Servers eingelegt und bietet die Möglichkeit menügesteuert die anschließende Betriebssysteminstallation vorzukonfigurieren. Dabei können alle Einstellungen bereits getroffen werden, die ansonsten während des Installationsprozesses eingegeben werden müssen.

- OS-Version: Windows Server 2003 R2 Standard
- Größe der Systempartition: 25 GB
- Name, Firma; KVMYK, Kreisverwaltung Mayen-Koblenz
- Lizenz-Nr. AABBC-C1122-3344E-E5566-FF77G
- Lizenzierung: pro Arbeitsplatz
- SNMP-Installation: ja und Trap-Destination: 172.16.10.130

"HP SmartStart" kann automatisch das SNMP-Protokoll sowie die Agenten zur Hardwareüberwachung des Servers, die "HP Insight Agents" installieren. Als Trap-Destination für SNMP wird die IP-Adresse 172.16.10.130 eingegeben, die den SNMP-Manager darstellt. Der "HP Systems Insight Manager" fungiert dann als zentrale Managementkomponente, überwacht den Status der Server und kann von dort gezielt Benachrichtigungen an Administratoren ausgeben.

Der Modus der Lizenzierung ist eine Besonderheit des Serverbetriebssystems. Eine Windows Workstation lässt immer nur maximal 10 gleichzeitige Verbindungen auf den Rechner zu. Diese Limitierung besteht beim Server nicht. Es gibt dort zwei unterschiedliche Möglichkeiten der Zählung der Lizenzen. Bei der Einstellung "pro Arbeitsplatz" wird die Gesamtzahl aller Arbeitsplätze gezählt, unabhängig davon, auf wie viele Server sie gleichzeitig zugreifen. Bei der Lizenzierung "pro Server" werden die Lizenzen fest einem Server zugeordnet. Der Lizenzmodus "pro Server" ist i. d. R. nur bei ganz kleinen Netzwerken mit einem oder zwei Servern sinnvoll. In den meisten Fällen ist es sinnvoller "pro Arbeitsplatz" zu lizenzieren. Bei einer Lizenzierung "pro Arbeitsplatz" wird nach einem Lizenzserver in der Organisation gesucht. Dort werden die Lizenzen zentral verwaltet und überwacht.

Danach muss nach Aufforderung die Windows Server 2003 R2 Standard CD eingelegt werden und die Installation läuft weitgehend automatisch. Lediglich das Administratorkennwort und der Rechnername müssen manuell eingegeben werden.

Bezüglich Fehlerbehebungen und Updates wird anschließend die Version *Srv03postSP2update-071212.exe* bzw. die Hotfixes kb899409 und kb919614 von Microsoft aufgespielt. Diese beheben Timingprobleme in Verbindung mit Terminaldiensten und sind nur auf gezielte Anforderung von Microsoft erhältlich.

Die Server sind über einen automatischen Umschalter an einen TFT-Monitor angeschlossen, der mit einer Auflösung von 1.280x1.024 Bildpunkten bei 60Hz betrieben wird. Im Folgenden werden nur noch die Unterschiede zum Windows XP Image erwähnt und so eine unnötige Wiederholung vermieden. Anschließend erfolgt die Installation der Windows Support Tools von der Windows Server 2003 CD.

Die Partitionierung der Festplatte erfolgt hier ähnlich zur Workstation, also Laufwerk C: für das Betriebssystem und Standardanwendungen mit einer Größe von 25 GB. Der Rest steht als Laufwerk D: zur Verfügung und beinhaltet das Verzeichnis *SPOOL* und *Temp*. Das Verzeichnis nimmt die temporären Dateien auf, die während des Spool-Vorgangs beim Drucken entstehen. Das Laufwerk E: entspricht dem DVD-ROM Laufwerk.

Gerätespezifische Treiber werden ebenfalls wie beim Windows XP Image eingespielt, allerdings werden keine Druckertreiber installiert. Gleiches gilt auch für die Standardanwendungen. Diese werden nicht installiert, sondern - wie die Druckertreiber auch - über den Citrix Installationmanager automatisch verteilt bzw. über die Citrix Presentation Server Konsole repliziert.

Die Konfiguration von "Automatische Updates" und der "Firewall" ist analog zu XP. Im Weiteren wird auf dem Server die Komponente "Verstärkte Sicherheitskonfiguration für den Internet Explorer" deinstalliert. Diese existiert nur auf dem Serverbetriebssystem und ist per Default aktiviert. Da im Zusammenhang mit der Benutzung von *sysprep.exe* Probleme bekannt sind, dass sich diese Deinstallation nicht auf die späteren Benutzer auswirken kann, ist präventiv ein zusätzlicher Eintrag in der *sysprep.inf* erforderlich. [KB908784]

Abschnitt [Unattended]

UpdateServerProfileDirectory=0	Verhindert mögliche Probleme mit der "Verstärkten Sicherheitskonfiguration"
KeepPageFile = 1	Legt das Verhalten der *pagefile.sys* fest; 0 = berechnet die pagefile.sys neu in Abhängigkeit der Speichergröße; 1 = die Größe bleibt unverändert

Im Unterschied zu einem Endgerät wird bei einer Terminalserverinstallation die Größe der Auslagerungsdatei *pagefile.sys* fest auf 2 GB eingestellt. Damit diese Einstellung erhalten bleibt, wird in der *sysprep.inf* ein weiterer Eintrag gesetzt.

Als weitere Komponente werden die "Terminaldienste" installiert, die gleichnamige Funktionen zur Verfügung stellen und die die Basis der späteren Installation des Citrix Presentation Servers darstellen. Im Zuge dessen werden auch die Zugriffe über das RDP-Protokoll aktiviert. Dazu wird für die Administratoren der Zugriff auf den Remotedesktop aktiviert, der später zur Fernwartung und Administration genutzt werden kann. Die Fernwartungssoftware NSM Netsupport Manager wird nicht installiert. Die Benutzer des Terminalservers greifen ausschließlich über das ICA-Protokoll auf den Server zu. Das ICA-Protokoll ist eine Entwicklung von Citrix und das Pendant zum RDP-Protokoll von Microsoft. Es ist allerdings wesentlich leistungsfähiger, kleiner und flexibler als das Microsoft Protokoll und bietet dar-

über hinaus Zugriff auf lokale Geräte des Thin-Clients, wie z.B. CD-ROM oder USB-Kartenleser.

Das Verzeichnis *c:\install* wird ebenfalls angelegt; allerdings werden dort nur die Dateien für die spätere Installation des Citrix Presentation Servers abgelegt. Damit die Installation später einwandfrei durchläuft muss der Datenbankzugriff auf den Citrix IMA-Dienst vorkonfiguriert hinterlegt werden. Dazu wird die Datei *IMASQL.dsn* in den Ordner *Data Sources* unterhalb von *c:\programme\Gemeinsame Dateien\ODBC* gelegt.

Der Vorgang der Systemvorbereitung mittels *sysprep.exe* entspricht dem bei der Installation von Windows XP. Wichtig ist hierbei, dass die entsprechende Version von *sysprep.exe* benutzt wird, die zum Server 2003 R2 gehört. Diese findet sich auf dem Datenträger und besitzt die Version "5.2.3790.1830 (srv03_sp1_rtm.050324-1447)". Für Windows XP SP2 lautet die Version "5.1.2600.2180 (xpsp_sp2_rtm.040803-2158)". Da die Serverversion von *sysprep.exe* neuer ist, kann diese auch für die Workstation benutzt werden, aber nicht umgekehrt. Natürlich müssen hier die entsprechenden Abschnitte der zugehörigen Datei *sysprep.inf* auf die Serverinstallation angepasst werden. Eine Besonderheit ist die bereits oben erwähnte Lizenzierung. Dafür gibt es in der Steuerdatei einen eigenen Eintrag namens "AutoMode" im Abschnitt "LicenseFilePrintData".

Abschnitt [LicenseFilePrintData]

AutoMode=PerSeat	Legt den Lizenzierungsmodus fest; PerSeat = Pro Client bzw. Arbeitsplatz PerServer = Pro Server

Ein Beispiel einer vollständigen *sysprep.inf* findet sich im Anhang "Beispieldateien sysprep.inf".

Die beiden Scriptdateien *1command.cmd* und *2command.cmd* entsprechen der Workstation Installation von XP. Die *1login.cmd* hat folgendes Aussehen:

```
@ECHO OFF
(1)@start /wait %InstDir%\Sysprep\dsmodts.cmd
(2)@start /wait %InstDir%\session\regtlb \\DBSRV8\Session$\SmcServer.tlb
(3)@start /wait %InstDir%\Citrix\mps_setup.cmd
(4)@start %InstDir%\Sysprep\shutdown.exe /L /R /T:10 "Reboot laeuft" /C
(5)start C:\install\Sysprep\2.cmd
```

Das Script *dsmodts.cmd* (1) nimmt den Terminalserver in eine bestimmte Gruppe im Active Directory auf, die dafür sorgt, dass die notwendigen Richtlinien und Berechtigungen für den Terminalserver erteilt werden. Der Abschnitt (2) ist von XP bekannt und entspricht dem dortigen Eintrag (6), ebenso wie der anschließende Neustart (4) und das Löschen der Scriptdateien (5). In der Zeile (3) erfolgt die automatisierte Installation von Citrix Presentation Server über den Aufruf einer separaten Batchdatei.

Natürlich unterscheidet sich der Server auch in der Art und Weise, wieder Name des Rechners gebildet wird. Dies wird durch eine INI-Datei gesteuert und ist im Abschnitt "Automatisierung der Imageinstallation" und "Scripting des Endgerätenamens" näher erläutert. Diese beiden Batchdateien sowie sonstige INI-Dateien sind als Beispiele im Anhang aufgeführt.

7.5 Bereitstellung der Images über PXE

Mittels PXE ist es für Computer möglich, auch ohne lokale Datenträger ins Netzwerk zu starten. Dies wird durch eine Erweiterung der Netzwerkkarte erreicht, die ein eigenes BIOS enthält. Mit dieser Technik sind Endgeräte realisierbar, die kein lokal installiertes Betriebssystem besitzen, sondern dieses bei jedem Start aus dem Netzwerk laden. Auch für ältere Rechner, die noch keine Netzwerkkarte mit PXE-Unterstützung haben, ist ein PXE-Bootvorgang über eine Diskette möglich. Vorwiegend wird PXE aber genutzt, um die Installation von Geräten zu vereinfachen.
Der generelle PXE-Boot-Ablauf sieht folgendermaßen aus und wird durch das Drücken der Taste "F12" nach dem Einschalten des Computers gestartet. Voraussetzung ist, dass die Funktion im BIOS des Rechners aktiviert ist.

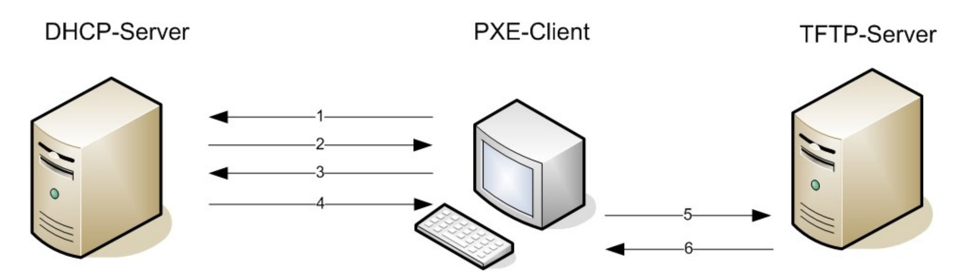


Abbildung 7.21 Normaler PXE-Bootvorgang

1. Broadcast DHCP Discover
2. Broadcast DHCP Offer
3. Broadcast DHCP Request
4. Broadcast DHCP ACK
5. TFTP Server (Option 66 "Hostname des TFTP-Startservers") anfragen
6. vom TFTP Server (Option 67 "Name der Startdatei") *pxelinux.0* herunterladen

Voraussetzungen für den Betrieb von PXE sind ein entsprechend konfigurierter DHCP-Dienst sowie ein TFTP-Server. Nach dem Einleiten des PXE-Startvorgangs sendet der PXE-Client einen DHCP-Broadcast (1) ab, der u. a. auch Informationen zu einem PXE-Server erwartet. Im Verlaufe der erfolgreichen Kommunikation (2-4) erhält der PXE-Client die benötigten Informationen und startet von dem ihm genannten TFTP-Server den Bootloader *pxelinux.0* (5-6). Der Vorgang ist am Beispiel des RIS-Servers aus Kapitel "Installation mit Microsoft RIS" noch ausführlicher dargestellt.

Der Bootloader *pxelinux.0* ist im Falle einer Thin-Client Installation ein Installationsprogramm, das aus Kapitel "Imageserver für Thin-Clients (Scout NG)" bekannt ist. Durch das Drücken der Taste "A" mit anschließendem "Return" wird die Installation gestartet.

Soll ein Fat-Client installiert werden, so wird der Bootloader vom Server IMAGESRV gestartet. Der Bootloader wird komfortabel mithilfe des 3Com Boot Image File Editors konfiguriert. Dieser ist Teil der 3ComBoot Services, die in einer OEM-Version auf der Ghost-CD zur Verfügung gestellt werden. Damit ist es möglich, den Bootloader als Menü auszuführen, der dann die Möglichkeit bietet entsprechende Vorgänge auszuführen. Über den Menüpunkt "Create a PXE menu boot file" kann dann über "Add" ein neuer Eintrag hinzugefügt werden und das gewünschte "Image file" ausgewählt werden.

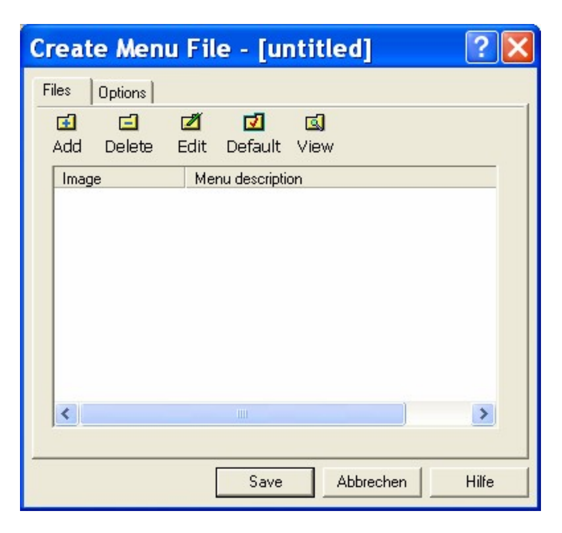


Abbildung 7.22 3Com Boot Image File Editor

Mit "Image file" ist hier allerdings nicht das zu verteilende Ghost-Image gemeint, sondern der Bootloader, der im Rahmen von PXE gestartet werden soll. Hier ist es sinnvoll, zunächst den Vorgang "Von lokaler Festplatte starten" zu erstellen. Wird der Haken bei dem Feld "Boot from hard drive" gesetzt, wird automatisch das Feld "Image file" inaktiv. Startet später aus Versehen ein Benutzer den PXE-Bootvorgang, startet der Rechner nach 5 Sekunden wie gewohnt von der lokalen Festplatte. Der Wert von 5 Sekunden wird unter dem Menüpunkt "Options" hinterlegt, genauso wie der Titel des Menüs "Auswahlmenu". Zu beachten ist, dass der Editor keine Umlaute und Sonderzeichen beherrscht.



Abbildung 7.23 3Com Boot Menu – erster Eintrag

Ein nachträgliches Ändern der Reihenfolge der Einträge ist nicht mehr möglich, deshalb sollte der Eintrag "Starten von Festplatte" als erstes erstellt werden und als Default hinterlegt werden. Dann sollte das Menü mit dem Namen *pxelinux.0* gespeichert werden. Es ist darauf zu achten, dass die Datei von Hand umbenannt werden muss, da das Programm beim Speichervorgang immer noch die Endung *mnu* an den Dateinamen hängt.

Die restlichen Einträge werden durch Auswahl eines "Image file" und entsprechender Beschreibung hinterlegt. Sinnvoll sind Einträge für das Erstellen, Zurückspielen und manuelle Starten von Ghost, d.h. ohne besondere Parameter. Im Weiteren sind hier noch BIOS-Updates und die automatische Einstellung des BIOS hinterlegt, die ebenfalls über PXE gestartet werden können.

Das fertige Menü könnte dann z.B. wie folgt aussehen.

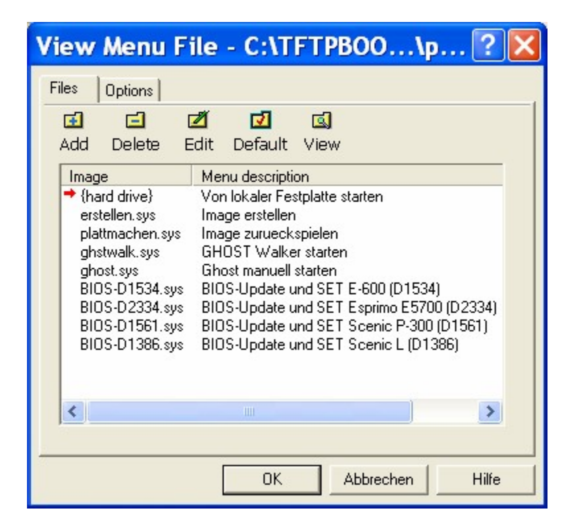


Abbildung 7.24 3Com Boot Menu – komplett

Hinter dem Auswahlpunkt "Image erstellen" verbirgt sich eine Imagedatei namens *erstellen.sys*. Diese entspricht einer Netzwerkbootdiskette, die über den Punkt "Network Boot Package" mithilfe des Symantec Ghost Boot Wizard erstellt wurde. Anschließend wurden die Parameter für den Aufruf von Ghost angepasst und die fertige Diskette mithilfe des Programms *rawritewin* in ein Image namens *erstellen.sys* konvertiert. Auf gleiche Weise wurde mit den Einträgen, die Ghost betreffen, vorgegangen. Bei dem Eintrag "GHOST Walker starten" wird eine DOS-Diskette gestartet, die die *ghstwlk.exe* automatisch startet. War ein Rechner bereits einmal Mitglied einer Domäne und wird von ihm ein Image erstellt, das auf andere Rechner verteilt werden soll, dann muss unbedingt die SID der Rechner vor dem Beitritt zur Domäne geändert werden. Nur die SID ist das entscheidende interne Identifikationsmerkmal eines Rechners in einer Domäne. Diese Änderung kann beispielsweise mit dem GHOST Walker oder anderen Programmen[29] absolviert werden. Diese Vorgehensweise war früher weit verbreitet, daher existiert der Eintrag noch, auch wenn er heute selten Verwendung findet.
Für den Anwender, der PXE auswählt, sieht das Menu dann folgendermaßen aus.



Abbildung 7.25 3Com Boot Menu – aus Anwendersicht

Die Erstellung des Images für das automatische BIOS-Update sowie die gleichzeitige Einstellung des BIOS ist ein wenig trickreich.

[29] Z.B. NewSID von Sysinternals http://technet.microsoft.com/en-us/sysinternals/bb897418.aspx

7.6 Automatisches BIOS per PXE

Die neueren Rechner haben im Regelfall kein Diskettenlaufwerk mehr. Die Hersteller haben darauf reagiert und bieten inzwischen BIOS-Updates auch per ISO-Datei an. Damit kann eine bootfähige CD erstellt werden. Für Rechner, die weder Diskettenlaufwerk noch ein optisches Medium besitzen, besteht bei FSC die Möglichkeit ein Update des BIOS im laufenden Betrieb unter Windows durchzuführen. HP bietet als Alternative den Weg über USB-Medien, wie z.B. einen USB-Stick. Hier stehen also genug Möglichkeiten ein Update durchzuführen zur Wahl.
Soll eine vorhandene Konfiguration, also die Einstellungen des BIOS, gesichert werden, so ist ein schreibender Zugriff notwendig. Ein Großteil der Endgeräte verfügt zudem über keine Windows-Umgebung mit direktem Zugriff auf das BIOS. Es scheidet also jegliche Automatisierung unter einem grafischen System aus. Daher bleibt nur der Weg über die Diskette bzw. über eine DOS-Umgebung mit Schreibzugriff übrig. Der Vorteil ist auch eindeutig die Geschwindigkeit, mit der ein DOS-System über das Netzwerk gestartet ist. Bei FSC geschieht das Update des BIOS z.B. über die *bioflash.exe*. Mit dem gleichem Programm können auch die Einstellungen des BIOS gesichert werden. Dazu wird das Programm mit dem Parameter "/CMOS" aufgerufen und die Einstellungen werden in eine Datei mit der Endung *omf* gesichert. Bei einem BIOS-Update wird immer nach der zugehörigen Datei mit der Endung *omf* gesucht. Wird sie gefunden, dann werden diese Einstellungen automatisch auf das BIOS des Rechners angewendet.
Laut den Vorgaben booten die Endgeräte als erstes über die lokale Festplatte und als zweites über PXE. Alle anderen Bootmedien sind deaktiviert. Wie eingangs erwähnt, besitzen die Geräte kein Diskettenlaufwerk. Es bleiben nur diese beiden Wege übrig, um das Update durchzuführen und die Einstellungen des BIOS in eine Datei zu sichern. Änderungen am BIOS des Rechners würden sich in der später erstellten Konfigurationsdatei des BIOS niederschlagen und somit in der Folge alle damit konfigurierten Rechner mit falschen Einstellungen beliefern. Als Beispiel sei der vorübergehende Einbau eines Diskettenlaufwerks genannt. Im BIOS müsste die Anwesenheit eines Laufwerks aktiviert und die Bootreihenfolge verändert werden, erst dann kann die *omf*-Datei erstellt werden. Alle zukünftigen Rechner haben dann im BIOS ein falsche Reihenfolge und ein Diskettenlaufwerk, das tatsächlich nicht vorhanden ist. Diese Fehlkonfiguration kann zu gravierenden Fehlern führen, ein Beispiel ist zu Anfang im Kapitel "Technische Arbeiten" genannt. Es bleibt nur der Weg über PXE und den lokalen Datenträger Festplatte übrig.

Um alle verschiedenen Rechnertypen mit ein und derselben Lösung zu bedienen, wird auf einem beliebigen Rechner auf die Festplatte ein minimales DOS mit Netzwerkunterstützung installiert. Dabei sind die Netzwerktreiber für sämtliche Modelle zu berücksichtigen. Weiterhin sollten alle aktuellen BIOS-Updates enthalten sein. In dieser Installation kann ein kleines Startmenü eingerichtet werden, wo das entsprechende Rechnermodell ausgewählt werden kann. In der Folge wird dann der korrekte Treiber geladen und ein Netzlaufwerk verbunden. Dorthin kann später die *omf*-Datei gesichert werden. Von dieser Installation wird dann ein Ghost-Image erstellt. Dieses kann dann bei Bedarf per PXE auf einen korrekt konfigurierten Rechner gespielt werden. Der Rechner kann das DOS anschließend von der lokalen Festplatte starten, das BIOS-Update durchführen und die Einstellungen in die *omf*-Datei sichern. Diese wird dann in das Netzlaufwerk gesichert.
Nun kann die Originaldiskette des Herstellers zusätzlich mit der *omf*-Datei bestückt und der Aufruf des BIOS-Updates mit den zwei Parametern "/AUTO" und "/Y" so automatisiert wer-

den, dass keine Benutzereingabe erforderlich ist. Nach dem Update führt der Rechner automatisch einen Neustart durch. Nun wird mit *rawritewin* ein Image dieser Diskette erstellt. Dieses wird dann - wie bereits gezeigt - in das PXE-Startmenü eingebaut, z.B. *BIOS-D1534.sys* mit dem Eintrag "BIOS-Update und SET E-600 (D1534)".

7.7 Automatisierung der Imageinstallation

Die Automatisierung der Imageinstallation lässt sich grob in vier Phasen einteilen. Dies wird hier anhand der Endgeräte ausgeführt, gilt aber in gleicher Weise für die Installation der Terminalserver. Unterschiede der beiden Installationen werden dargestellt.

- Die Phase 1 beinhaltet alle Maßnahmen, die den Übergang vom Basisimage *PC60xyzversion.gho* zum Masterimage *PC60xyzversion_sysprep.gho* kennzeichnen.
- Phase 2 wird hier als "Pre-Installation"-Task bezeichnet und meint alles, was vor dem Start des eigentlichen System-Preparation-Process mit *sysprep.exe* von Microsoft geschieht.
- Die Phase 3 bezeichnet dann den vollautomatisierbaren Vorgang mittels *sysprep.exe*.
- Die Phase 4 bezeichnet alle Vorgänge nach dem Abschluss des Vorgangs mit *sysprep.exe*.

Alle Phasen hängen eng zusammen, die Unterteilung macht es aber leichter die einzelnen Abschnitte zu erklären und dabei den Überblick zu behalten.

Die Vollautomatisierung wird durch fertige und frei verfügbare Tools und selbstgeschriebene Scripte erreicht, die durch diverse Batchdateien aufgerufen und gesteuert werden. Soweit im Folgenden Batchdateien erläutert werden, ist immer nur der relevante Teil des Scripts dargestellt. Die kompletten Scripts finden sich im Anhang und enthalten in den meisten Fällen noch Fehlersteuerung und Bildschirmausgaben.

Phase 1

Sie setzt direkt auf dem fertigen Basisimage auf, beispielsweise *PC60xyz23.gho*. Um von dem Basisimage zu dem Masterimage *PC60xyz23_sysprep.gho* zu kommen, das dann später verteilt werden soll, müssen folgende Schritte und Einstellungen auf dem vorgenannten Basisimage vorgenommen werden:

1. Den Schlüssel "RunOnce" in der Registry auf folgenden Wert setzen.
   ```
   [HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce]
   @="C:\\Sysprep\\ip2name.cmd"
   ```
2. (optional) Das lokale Administratorkennwort auf einen definierten Wert setzen, z.B. mittels des Programms *pspasswd.exe*.
   ```
   C:\sysprep\pspasswd administrator password
   ```
3. Den Autologin aktivieren, z.B. mittels *autologon.exe*.
   ```
   C:\Sysprep\Autologon\autologon administrator RechnerName password
   ```
4. Die DHCP-Adresse des Rechners freigeben, z.B. mittels *ipconfig*.
   ```
   ipconfig /release
   ```
5. Den Rechner neu starten und das Image *PC60xyz23_sysprep.gho* erstellen.

Das Ganze sieht in einer kleinen Stapelverarbeitungsdatei namens *autologin.cmd* so aus:

```
:SetRunOnce
reg import C:\Sysprep\autologon\RunOnce_ip2name.reg
:SetDefaultPwd
CALL C:\sysprep\pspasswd administrator password
:AutoLoginOn
C:\Sysprep\Autologon\autologon administrator pc60xyz password
```

```
:ReleaseIp
Ipconfig /release
```

Unmittelbar nach der Freigabe der DHCP-Adresse sollte der Rechner neu gestartet und das Image erstellt werden. So wird gewährleistet, dass das Image keine IP-Adresse besitzt und beim nächsten Start einen DHCP-Broadcast absetzt, um eine gültige IP-Adresse zu bekommen. Theoretisch ist nicht notwendig, da beim ordnungsgemäßen Herunterfahren des Betriebssystems die IP-Adresse freigegeben werden sollte. Aus Erfahrung lässt sich aber sagen, dass das nicht immer der Fall ist. Der Rechner speichert die letzte Adresse zwischen (cached sozusagen) und startet auch wieder mit dieser Adresse. So kann es im ungünstigsten Fall zu einem Konflikt der IP-Adressen kommen, wenn zwei oder mehr Rechner gleichzeitig von diesem Image installiert werden. Mittels des Freigebens der IP-Adresse ist man auf der sicheren Seite und vermeidet von vorneherein diesen Fall.

Phase 2

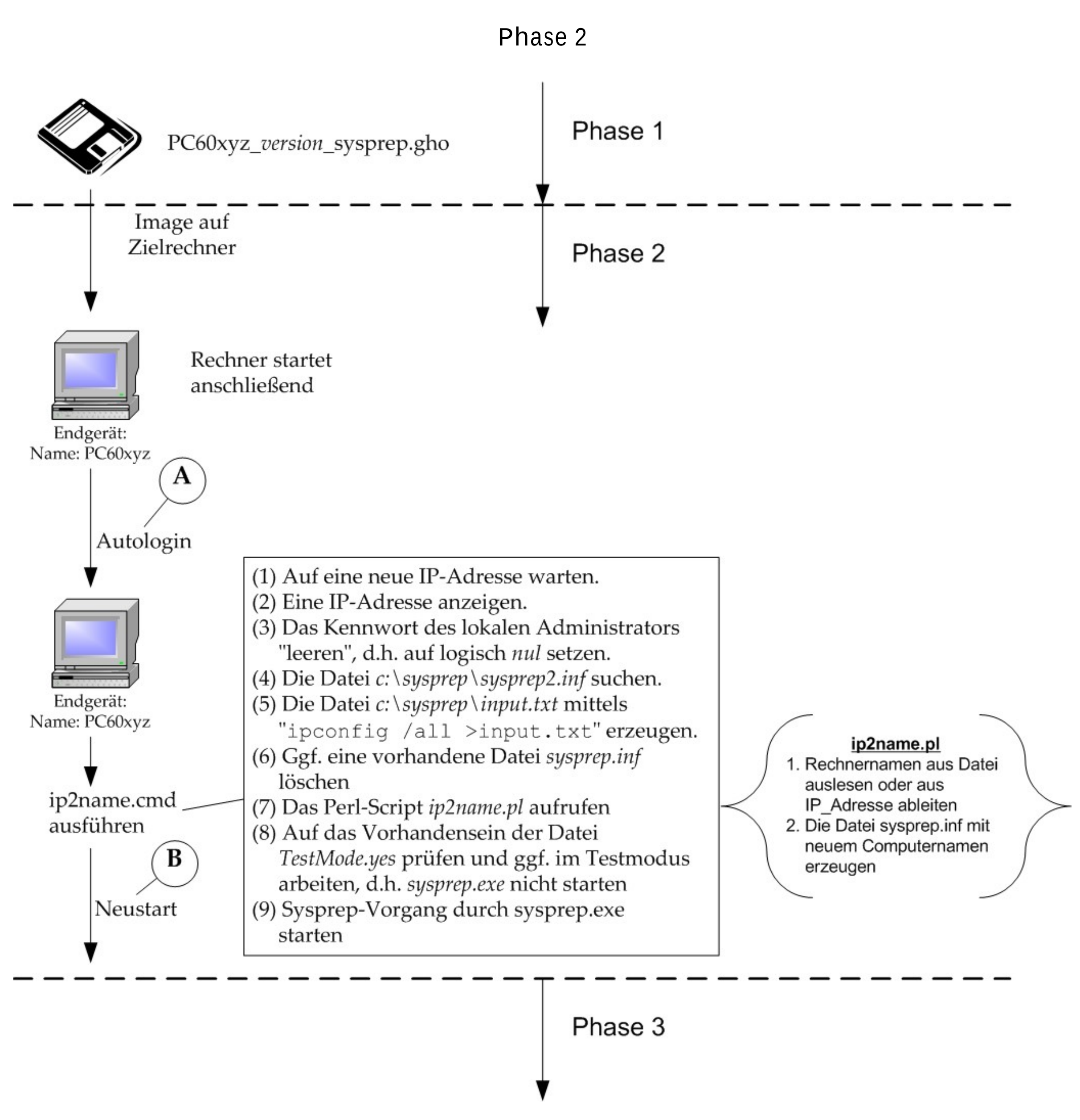


Abbildung 7.26 Phase 1 und 2 der automatischen Installation

Wird das Image *PC60xyz23_sysprep.gho* auf einen Rechner zurückgespielt, startet der Rechner anschließend automatisch neu. In den beiden Zeichnungen zu Phase 1 und 2 bzw. Phase 3 und 4 sind markante Positionen mit den großen Buchstaben A-E gekennzeichnet. In den Erläuterungen werden diese z.B. durch "A)" kenntlich gemacht.
Mit dem automatischen Login als lokaler Administrator startet dann die Phase 2. In der nun folgenden Aufzählung sind die einzelnen Vorgänge in Schritte und Unterschritte unterteilt.

A) Automatischer Login und Start des Batch-Scripts *c:\sysprep\ip2name.cmd*.
  (1) Auf eine neue IP-Adresse warten. (:WaitForIp)
  (2) Eine IP-Adresse anzeigen. (:IPok)
  (3) Das Kennwort des lokalen Administrators "leeren", d.h. auf logisch *nul* setzen. (:ResetPwd)
  (4) Die Datei *c:\sysprep\sysprep2.inf* suchen. (:SrchSysprep2)
  (5) Die Datei *c:\sysprep\input.txt* mittels "`ipconfig /all >input.txt`" erzeugen. (:Input)
  (6) Ggf. eine vorhandene Datei *sysprep.inf* löschen (:Inputok)
  (7) Das Perl-Script *ip2name.pl* aufrufen (:StartScript)
  (8) Auf das Vorhandensein der Datei *TestMode.yes* prüfen und ggf. im Testmodus arbeiten, d.h. *sysprep.exe* nicht starten (:CheckTestMode)
  (9) Sysprep-Vorgang starten (:MakeSysprep)

Das Script wurde auch da ausführlich mit Labels gekennzeichnet, wo es nicht notwendig war, um die Lesbarkeit zu verbessern. Die Labels beginnen mit dem Doppelpunkt und sind in der Auflistung in Klammern dahinter angefügt. Weiterhin sind die Zeilen stellenweise mit Zahlen versehen, um die Erläuterung zu vereinfachen. Hier ist als Beispiel das oben genannte Script *ip2name.cmd* auszugsweise dargestellt:

```
(1)  SET SysprepDir=c:\sysprep
(2)  ipconfig /renew >nul
     :WaitForIp
(3)  %SysprepDir%\wait 10 >nul
(4)  ipconfig >%SysprepDir%\connect
(5)  type %SysprepDir%\connect | find "kein Vorgang" 2>&1 >nul
(6)  IF %ErrorLevel% == 0 ECHO -* Netzwerkkabel nicht gesteckt...
(7)  IF %ErrorLevel% == 0 GOTO WaitForIp
(8)  type %SysprepDir%\connect | find "IP-Konfiguration" 2>&1 >nul
(9)  IF %ErrorLevel% == 0 GOTO CheckIp
(10) GOTO WaitForIp
     :CheckIp
(11) type %SysprepDir%\connect | find "IP-Adresse" 2>&1 >nul
(12) IF %ErrorLevel% == 1 ECHO -* noch keine IP-Adresse erhalten
(13) IF %ErrorLevel% == 1 GOTO WaitForIp
(14) IF %ErrorLevel% == 0 GOTO IPok
     GOTO WaitForIp
     :IPok
(15) IPCONFIG | find "IP-Adresse" 2>&1
     :SrchSysprep2
(16) IF EXIST %SysprepDir%\sysprep2.inf GOTO Input
     GOTO ENDE
     :Input
(17) IPCONFIG /all>%SysprepDir%\input.txt
     :INPUTOK
(18) IF EXIST %SysprepDir%\Sysprep.inf del %SysprepDir%\sysprep.inf >nul
     :StartScript
(19) %SysprepDir%\perl\bin\perl %SysprepDir%\ip2name.pl
     :CheckTestMode
(20) IF EXIST %SysprepDir%\Testmode.yes GOTO Testbetrieb >nul
     :ResetPwd
(21) CALL C:\sysprep\pspasswd administrator >nul
```

```
      :MakeSysprep
 (22) %SysprepDir%\sysprep -mini -activated -quiet -reseal –reboot
      GOTO ENDE
      :Testbetrieb
```

(1) Das Script setzt am Start eine Variable namens "SysprepDir", der der Wert "c:\sysprep" zugewiesen wird. Zugriff auf den Inhalt der Variablen erhält man über %SysprepDir%.

(2) Mit dem Befehl *ipconfig /release* wird die IP-Adresse des Adapters freigegeben.

(3) Das kleine Tool *wait.exe* wartet anschließend die angegebene Zeit in Sekunden und gibt dabei akustische Signale über den Lautsprecher aus.

(4) Mit dem Befehl *ipconfig* wird die TCP/IP-Konfiguration in eine kleine Textdatei namens *connect* umgeleitet.

(5) Anschließend versucht das Betriebssystemprogramm *find* den String "kein Vorrang" in dieser Datei zu finden. Dieser String erscheint nur, wenn kein Netzwerkkabel aufgesteckt ist.

(6) Im Anschluss daran wird der ErrorLevel des Vorgangs erfragt. Dabei wird bei Erfolg 0 sonst ein Fehlerwert > 0 zurückgegeben.

(7) Ist der Errorlevel 0, ist kein Kabel aufgesteckt und das Script springt wieder an die Stelle ":WaitForIp".

(8) Anschließend wird in der Datei *connect* der String "IP-Konfiguration" gesucht, um sicherzustellen, dass auch TCP/IP korrekt initialisiert worden ist.

(9) War die Suche erfolgreich, wird zur Sprungmarke ":CheckIp" verzweigt.

(10) Im Fehlerfall springt das Script wieder zu ":WaitForIp".

(11) An dieser Stelle wird der String "IP-Adresse" mit *find* in der Datei *connect* gesucht.

(12-13) Im Fehlerfall wird eine Meldung ausgegeben und wieder zu ":WaitForIp" verzweigt.

(14) Bei Erfolg wird zur Marke ":IPok" verzweigt.

(15) Diese Zeile dient nur dazu, die IP-Adresse am Bildschirm anzuzeigen. Der "Erfolg" der *find*-Befehls ist bereits bei (:CheckIp) sichergestellt worden.

(16) Anschließend wird nach der Existenz der Datei *sysprep2.inf* geschaut und zur Marke ":Input" gesprungen, ansonsten wird das Script vorzeitig verlassen. Diese Datei ist quasi die Vorlage für die später zu erstellende Datei *sysprep.inf*.

(17) An dieser Stelle wird die Datei *input.txt* durch Umleiten der Ausgabe von *ipconfig /all* erzeugt. Die Datei enthält nun sowohl TCP/IP-Informationen der angeschlossenen Adapter, als auch IP- und MAC-Adressen.

(18) Eine eventuell schon vorhandene Datei *sysprep.inf* wird gelöscht.

(19) Sind nun alle Voraussetzungen gegeben, kann das Perl-Script *ip2name.pl* gestartet werden. Dieses Script hat in der Basisversion die IP-Adresse des Rechners bestimmt und aus diversen Vorgaben und der IP-Adresse den Rechnernamen erstellt. Dieser erzeugte Rechnername wird dann zur Vervollständigung der Datei *sysprep.inf* benutzt. Dazu wird die Vorlage *sysprep2.inf* eingelesen, die den leeren Eintrag "Computername=" enthält. Hinter den vorgenannten Eintrag wird der erzeugte Computername gesetzt und anschließend die Steuerdatei *c:\sysprep\sysprep2.inf* erzeugt. In der zurzeit vorliegenden Version 0.4 ist es auch möglich, eine zentrale Informationsdatei per http von einem Webserver herunterzuladen. Diese Informationsdatei enthält Wertepaare MAC-Adresse/Rechnername. Aus der oben erzeugten *input.txt* wird die MAC-Adresse extrahiert und in der Informationsdatei nach dem zugehörigen Rechnername gesucht. Ist die Suche erfolglos, kann bei aktiviertem Fallback-Modus auf die Erzeugung des Rechnernamens aus der IP-Adresse ausgewichen werden. Genaueres zu diesem Script findet sich in Kapitel 7.8 ff.

(20) Ist der Testmodus aktiv, d.h. die Datei *c:\sysprep\testmode.yes* vorhanden, wird abschließend die *sysprep.exe* nicht aufgerufen, sondern übersprungen.

(21) Um den Sysprep-Prozess mit anschließendem automatischem Login erfolgreich zu starten, darf der lokale Administrator zum Zeitpunkt des Starts von *sysprep.exe* kein Kennwort enthalten! Aus Sicherheitsgründen hat der Account natürlich immer ein Kennwort, es wird nur unmittelbar vor dem Aufruf von *sysprep.exe* entfernt. Es ist zu diesem Zeitpunkt aber immer noch möglich, das Script mit der Tastenkombination "Strg-C" abzubrechen. Um das auszuschließen, wird das Script nur bei gesperrtem Bildschirm gestartet. Dies ist auch im Anhang, wo das komplette Script abgedruckt ist, so gemacht. So ist indes keine optische Überwachung des Verlaufs des Scripts mehr möglich, allerdings kann diese Bildschirmsperre durch den Administrator jederzeit mit dem korrekten Kennwort aufgehoben werden.
(22) Nun kann der Prozess der Systemvorbereitungsdienste gestartet werden. Die Befehlszeilenparameter hinter *sysprep.exe* sind im Kapitel "Erstellung des Windows XP Images" erläutert, der Parameter "–reboot" bedeutet, dass der Rechner bei Beendigung des Sysprep-Prozesses automatisch neu gestartet wird.

Die Steuerdatei *sysprep.inf* ist für den Vorgang der automatischen Installation sehr wichtig und die für diese Phase relevanten Teile haben folgende Werte:

```
[BenutzerData]
    ProductKey= AAAAA-12345-BBBBB-67890-CCCCC
    FullName="KVMYK"
    OrgName="Kreisverwaltung Mayen-Koblenz"
    ComputerName=PC60114
```

Wir haben mit dem Script *ip2name.pl* die leere Stelle hinter den Eintrag "Computername=" gefüllt und können daher die Installation weiter automatisiert durchführen.

B) Automatischer Neustart nach der Initialisierung des Sysprep-Vorgangs.

Phase 3

C) Die folgende automatische Installation wird ohne Benutzereingriff durchgeführt und nur von der Datei *sysprep.inf* gesteuert. Die Datei *c:\sysprep\i386\oem\Cmdlines.txt* wird ausgewertet und die dort enthaltenen Scripts werden während der Miniinstallation ausgeführt. Die Datei enthält in unserem Fall zwei Scripts, *C:\Sysprep\1command.cmd* und *C:\Sysprep\2command.cmd*. Die *2command.cmd* wird zurzeit nicht benutzt, die *1command.cmd* führt einen wichtigen Befehl aus:
    (1) `@C:\Sysprep\Sysprep –clean`

Für diesen Phasenabschnitt sind folgende Wertezuordnungen aus der Steuerdatei *sysprep.inf* ausschlaggebend:

```
[Unattended]
    OemSkipEula=Yes
    InstallFilesPath=C:\sysprep\i386

[GuiUnattended]
 OEMSkipRegional=1
 TimeZone=110
 OemSkipWelcome=1

[Identification]
    JoinDomain=KV
    DomainAdmin=PCinst
    DomainAdminPassword=pcinst
    MachineObjectOU="OU = Computer,OU = KVMYK, DC = kv,DC = myk,DC = intern"


[sysprepcleanup]
Service=pci
```

```
Service=isapnp
…
```

Der Befehl `C:\Sysprep\Sysprep –clean` aus der *1command.com* bereinigt die Registry um die an der Stelle [sysprepcleanup] angegeben Treiber, wobei die Liste hier nur verkürzt dargestellt ist.
Bisher nicht erläutert wurde auch der Eintrag "MachineObjectOU". Hier wird hinterlegt, an welcher Stelle im Active Directory, genauer gesagt in welcher OU, das Computerobjekt abgelegt werden soll. Die Notation muss in LDAP-Schreibweise erfolgen und voll qualifiziert sein.

D) Neustart nach der automatischen Installation mittels Sysprep.

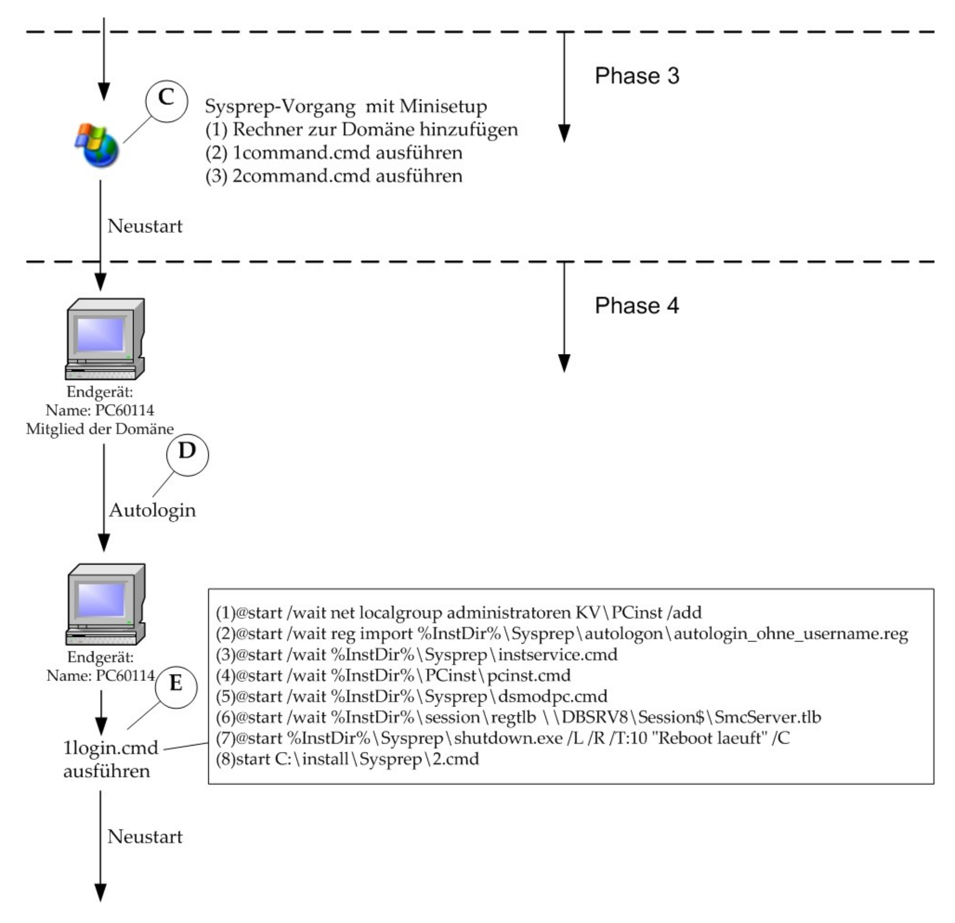


Abbildung 7.27 Phase 3 und 4 der automatischen Installation

Phase 4

Die Phase 4 ist gekennzeichnet durch Post-Image-Tasks, also Abläufen, die nichts mehr mit dem eigentlichen Vorgang des Rückspielens eines Images zu tun haben, sondern eine zusätzliche optionale Automatisierung im Rahmen des Systemvorbereitungsprozesses darstellen. Für den folgenden Phasenabschnitt E sind die folgenden Wertezuordnungen aus der Steuerdatei *sysprep.inf* ausschlaggebend:

```
[Unattended]
 OEMPnPDriversPath = "PnPDrvrs\E5915\Graphic\Graphics;PnPDrvrs\VMware\Graphic\"
 AutoLogon=Yes
 AutoLogonCount=1

[GuiUnattended]
```

```
    AdminPassword=administrator
    EncryptedAdminPassword=NO

   [GuiRunOnce]
    Command0=C:\Install\Sysprep\1login.cmd
```

E) Als Folge des Eintrages "`[GuiRunOnce] Command0=c:\install\sysprep\1login.cmd`" in der Datei *Sysprep.inf*, erfolgt ein automatischer Login (als lokaler Administrator) und Start des Batch-Scripts *1login.cmd*, welches schon im Kapitel "Erstellung des Windows XP Images" für Endgeräte ausführlich erläutert worden ist.

Eine der wichtigsten Einstellungen, die hier getroffen werden, ist zur Erhöhung der Benutzerfreundlichkeit gedacht. Dabei wird durch die Änderung einiger Registry-Schlüssel [`(2)… reg import…autologin_ohne_username.reg`] dafür gesorgt, dass nach dem nächsten Neustart die Anmeldemaske ein leeres Feld für den Benutzernamen enthält und die Domäne KV als Logon-Domäne vorgeblendet wird.



Abbildung 7.28 Anmeldemaske

Weiterhin wichtig ist das Hinzufügen des gerade installierten PCs zu einer definierten Gruppe im Active Directory [`(5)… C:\Install\Sysprep\dsmodpc.cmd`]. Erst die Mitgliedschaft in der Gruppe "bEndgeraeteMaschinen" sorgt dafür, dass der Rechner alle notwendigen Einstellungen über Richtlinien erhält. Terminalserver kommen natürlich über das Script *dsmodts.cmd* in die entsprechende Gruppe "bCitrixServer".

Sind alle Einstellungen erfolgt, sorgt die *2.cmd* dafür, dass alle Scripts, die Kennwörter enthalten, gelöscht werden. Auch hier wird im Anhang deutlich, dass bereits beim Start der *1login.cmd* der Bildschirm gesperrt wird, so dass es für den normalen Benutzer nicht möglich ist, die laufenden Scripte unrechtmäßig zu beenden, um somit administrative Rechte an dem Rechner zu erlangen. Abschließend wird der Rechner noch einmal neu gestartet und ist dann für den Endanwender einsatzbereit. Im Falle einer Terminalserverinstallation kann die Sperrung des Computers während der Installation ggf. unterbleiben.

7.8 Scripting des Endgerätenamens

Die Scriptsprache Perl hat den großen Vorteil, dass sie plattformunabhängig und modular aufgebaut ist. Ein Script läuft also unverändert auf einem Windows- oder Linux-Betriebssystem. Wichtige Voraussetzung ist dabei allerdings, dass nur Module Verwendung finden, die ebenfalls plattformunabhängig sind. Gleiches gilt auch für den Scriptteil, der selbst erstellt wird. Soweit möglich werden für das Script benötigte Informationen durch entsprechende Befehle des Betriebssystems zur Verfügung gestellt. Ein Beispiel dafür ist der

Befehl "ipconfig" von Windows, der Informationen zu TCP/IP liefern kann. Dieser wird im Rahmen der betriebssystemabhängigen Batchdatei *ip2name.cmd* aufgerufen. Soll das Perl-Script später auf einer Linux-Plattform betrieben werden, muss nur das Batch-Script *ip2name.cmd* auf die geänderte Umgebung angepasst werden. Das Perl-Script bleibt unverändert. Auch die erste Zeile des Perl-Scripts `#!/perl/bin/perl` verdeutlicht die Unabhängigkeit. Diese Zeile weist unter Linux auf den Pfad zum Perl-Interpreter hin, unter Windows wird diese Zeile komplett ignoriert.

Die folgenden Listings erleichtern das Verständnis von *ip2name.pl* und sind deshalb ausführlich dargestellt. Den Anfang macht die Datei *input.txt*, die durch den Aufruf `ipconfig -all >input.txt` gewonnen wird. Ohne den Parameter "-all" werden nur minimale IP-Informationen ausgegeben, mit Angabe des Parameters erscheinen auch die Informationen zur MAC-Adresse des Adapters. Siehe dazu einen Auszug aus der *input.txt*.

```
Windows-IP-Konfiguration

        Hostname. . . . . . . . . . . . . : PC50090
        Primäres DNS-Suffix . . . . . . . : kv.myk.intern
        Knotentyp . . . . . . . . . . . . : Hybrid
        IP-Routing aktiviert. . . . . . . : Nein
        WINS-Proxy aktiviert. . . . . . . : Nein
        DNS-Suffixsuchliste . . . . . . . : kv.myk.intern
                                            kv.myk.intern
                                            myk.intern

Ethernetadapter LAN-Verbindung:
        Verbindungsspezifisches DNS-Suffix: kv.myk.intern
        Beschreibung. . . . . . . . . . . : Realtek RTL8169/8110 Family Gigabit
Ethernet NIC
        Physikalische Adresse . . . . . . : 00-30-05-BA-67-14
        DHCP aktiviert. . . . . . . . . . : Ja
```

Eine typische *ip2name.txt* kann so aussehen.

```
11-22-33-44-55-66 TestingPcName; don't remove
00-30-05-45-67-3E PC123456; this is commentary
```

Eine minimale *sysprep2.inf* kann folgenden Inhalt haben.

```
[Unattended]
    OemSkipEula=Yes
    InstallFilesPath=C:\sysprep\i386
    TargetPath=\WINDOWS
    OEMPnPDriversPath = "PnPDrvrs\E5915\Graphic\;PnPDrvrs\VMware\Graphic\"
    OemPreinstall = Yes

[GuiUnattended]
    AdminPassword=password
    EncryptedAdminPassword=NO
    OEMSkipRegional=1
    OEMDuplicatorstring=PC60xyz-Baseimage-Sysprep-11122007
    TimeZone=110
    OemSkipWelcome=1
    AutoLogon=Yes
    AutoLogonCount=1

[BenutzerData]
    ProductKey=AAAAA-12345-BBBBB-67890-CCCCC
    FullName="FullName"
    OrgName="Organization"
    ComputerName=

[Identification]
    JoinDomain=DomainName
    DomainAdmin=AdminBenutzer
```

```
        DomainAdminPassword=password
        MachineObjectOU="OU = Standard,OU = Computer, DC = DomainName, DC = org"

    [Networking]

    [GuiRunOnce]
        Command0=C:\Install\Sysprep\1login.cmd
```

Die für das Perl-Script wichtigste Stelle ist "Computername=". Der Eintrag dahinter muss leer sein und wird erst durch das Script gefüllt. Die Datei ist dabei quasi die Vorlagendatei für die spätere *sysprep.inf*. Die Datei *sysprep2.inf* wird vom Script eingelesen, der spätere Name des Rechners wird an die richtige Stelle geschrieben und die Datei unter dem Namen *sysprep.inf* abgespeichert. Wie der Name des künftigen Rechners ermittelt wird, zeigen die nächsten Kapitel.

Unter Windows wird das Script mittels `perl.exe ip2name.pl` aufgerufen. Befinden sich der Perl-Interpreter und das Script nicht im gleichen Verzeichnis, muss der Pfad zum Interpreter mit eingegeben werden. Soll das Script aus einem Verzeichnis aufgerufen werden, in dem weder das Script selbst noch der Interpreter liegen, müssen beide mit vollem Pfadnamen aufgerufen werden, z.B. `c:\sysprep\perl\bin\perl.exe c:\sysprep\ip2name.pl`.

Im Normalbetrieb läuft das Script ohne Parameter durch. Die Steuerung von notwendigen Einstellungen und die Vorbelegung von Variablen erfolgt mithilfe einer Initialisierungsdatei oder kurz INI-Datei. Die INI-Datei hat die Endung *ini* und heißt *ip2name.ini*. INI-Dateien sind vergleichbar zu Konfigurationsdateien unter Linux, die oft die Endung *conf* besitzen. Der Aufbau der textbasierten INI-Datei ist generell wie folgt:

```
#Kommentar
[Sektion]
#Kommentar
Schlüssel=Wert
```

Das Script liest aus der INI-Datei in den entsprechenden Sektionen die erforderlichen Schlüssel aus und ordnet die gefundenen Werte Variablen im Script zu. Fehlen benötigte Werte, werden diese innerhalb des Scripts mit Default-Werten belegt. Eine grundlegende Sektion der *ip2name.ini* ist hier dargestellt.

```
[SOURCE]
# SysprepTpl: templ.File for building sysprep.inf
SysprepTpl=c:\sysprep\sysprep2.inf

# SysprepSrchStr: string to search for in sysprep2.inf
SysprepSrchStr=ComputerName=

[TARGET]
# Sysprep: must named sysprep.inf and placed in c:\sysprep
Sysprep=c:\sysprep\sysprep.inf

# Net: Class A, B or C
# Net=A results in HostMask=111111111
# Net=B results in HostMask=111111
# Net=C results in HostMask=111
Net=C

# HostMask must read from right to left
# indicates the bits used for building hostname
# use of HostMask will override Parameter Net
#
HostMask=

# HostMaskCheck: enable/disable minimum length of 3 for HostMask
# enable=1/disable=0
HostMaskCheck=1
```

```
# Suffix: first Part of the Computername
Suffix=PC

# crunches the resulting computername, if to long
CmpNameLength=15
```

Mit dem Schlüssel `SysprepTpl` wird die Vorlagendatei definiert, aus der später die Datei *sysprep.inf* generiert wird. Der Suchstring `SysprepSrchStr` verweist auf den String, der in der Vorlagendatei gefunden werden soll. Hinter diese Stelle wird später der neue Name des Rechners geschrieben.
In der Sektion `[TARGET]` stehen alle Werte, die das Ziel näher abgrenzen. Mit dem Schlüssel `Sysprep` wird das Ziel, die Erstellung der *sysprep.inf*, genau spezifiziert. Für den Prozess der Systemvorbereitung muss zwingend der Wert `c:\sysprep\sysprep.inf` verwendet werden. Zu Testzwecken kann der Wert aber durchaus abgeändert werden.
Der Parameter `Net` stellt eine vereinfachte Möglichkeit dar, den Schlüssel `HostMask` genau zu spezifizieren. Wird beispielsweise `Net=B` gesetzt, wird im Hintergrund automatisch der Wert von `HostMask=111111` gesetzt, ohne dass das explizit an anderer Stelle getan werden muss. Wird der Parameter `HostMask` allerdings gesetzt, wird der Wert von `Net` ignoriert. `HostMask` kann prinzipiell die Werte von 1 bis 111111111 annehmen, also eine einzige oder maximal 9 Einsen. Andere Werte sind nicht zulässig.
Der Schlüssel `HostMaskCheck` vom Typ boolean (true=1/false=0) regelt, ob `HostMask` alle erlaubten Werte annehmen darf oder mindestens den Wert 111 oder größer annehmen muss.
Im Schlüssel `Suffix` wird festgelegt, mit welchem String die Rechnernamen beginnen. Hier sei erneut darauf hingewiesen, welche Restriktionen für den Rechnernamen unter Windows gelten. [KB909264]
Zur Beschränkung der Länge des Computernamens dient der Wert des Schlüssels `CmpNameLength`. Hier gilt für Windows die Beschränkung auf 15 Zeichen.

Der Schlüssel `HostMask` wurde in Anlehnung an die bekannte SubnetMask so genannt und erfüllt hier eine ähnliche Funktion bei der Bildung des Computernamens. Bei der Bildung des Namens aus der ihm eigenen IP-Adresse werden die drei letzten Oktette der IP-Adresse auf je drei Stellen aufgefüllt. Dies erfolgt durch das Einfügen von führenden Nullen in jedem einzelnen Oktett. Die trennenden Punkte sowie das erste Oktett werden entfernt und die drei Oktette werden zu einem String namens Oktett3 zusammengefügt.
Zum besseren Verständnis dient folgendes Beispiel, das in tabellarischer Form einen besseren Überblick bietet:

IP-Adresse des Rechners	=	172.16.50.90
Oktett3	=	016050090
HostMask	=	11111
HostNamePart	=	50090
Suffix	=	PC
Resultierender Computername	=	PC50090

Tabelle 7.6 Einfluss der **HostMask** auf den Computernamen

Wer die Bedeutung der SubnetMask kennt, kann auch sehr schnell die Arbeitsweise der HostMask erkennen. Im Gegensatz zur SubnetMask wird die HostMask immer von rechts nach links ausgewertet und arbeitet mit dezimalen Werten. Nur die Teile des `Oktett3`, die einer Eins "zugeordnet" sind, bleiben erhalten, der Rest wird verworfen. In obigem Beispiel sind das die letzten 5 Stellen von `Oktett3`, die zusammen mit dem `Suffix=PC` und dem

`HostNamePart=50090` den resultierenden Namen *PC50090* ergeben. Eine ausführliche Betrachtung erfolgt im Kapitel "Definitionen", wo Einschränkungen und sinnvolle Werte für die `HostMask` besprochen werden, um die Konsistenz und Eindeutigkeit von Rechnernamen zu gewährleisten.

7.8.1 Definitionen

Ein Array wird durch ein vorangestelltes "@" kenntlich gemacht, z.B. "@ARGV". Der Zugriff auf einzelne Elemente des eben genannten Arrays erfolgt durch eine Integer-Zahl, die in eckigen Klammern angegeben wird und die Position des Elementes innerhalb des Arrays angibt. Das erste Element steht an der Stelle 0. Soll ein Zugriff auf das dritte Element erfolgen, geschieht dies durch "$ARGV[2]". Wird nicht das gesamte Array, sondern nur ein Element des Arrays angesprochen, wird das "@" durch ein "$" ersetzt. Das Array "@ARGV" enthält sämtliche Befehlszeilenparameter, die beim Aufruf des Scripts übergeben wurden. In Perl beginnen Variablen immer mit dem Zeichen "$", Unterprogramme werden mit einem führenden "&" aufgerufen. Perl unterscheidet zwischen Groß- und Kleinschreibung.
Kommentare werden durch das Zeichen "#" eingeleitet und reichen bis zum Ende der Zeile.

Das "eval"-Befehl bietet Möglichkeiten Fehler abzufangen. Ein Block wird durch geschweifte Klammern "{...}" eingegrenzt. Das eval-Konstrukt ähnelt dem "Try-and-Catch"-Block von Java. Mögliche Fehler werden in der Variable "$@" zurückgeliefert. Als Beispiel diene die mögliche Division durch Null.

```
eval {$ergebnis = $a / $b};
print $@ if $@;
```

Jede Befehlszeile endet mit einem Semikolon ";".
"StdOut" definiert die Standardausgabe, die im vorliegenden Fall die Ausgabe am Bildschirm ist. Wird eine Logdatei geführt, so ist es möglich "StdOut" und die Fehlermeldungen in die Logdatei umzuleiten. Diese kann dann z.B. zu Support-Zwecken per E-Mail versendet werden.

Die folgenden Variablen wurden innerhalb des Programms *ip2name.pl* benutzt. Weniger wichtige sind nicht aufgeführt, sondern nur solche, die zum Verständnis des Scripts erforderlich sind. Alle Variablen, die mit "$Vm" beginnen, werden nur verwendet, wenn das Script mit dem Parameter "-vm" aufgerufen wurde.

Name der Variable	Zweck	Erläuterung
$ARGV[i]	Das Array enthält die Befehlszeilenargumente.	Globales Spezialarray von Perl
$@	Enthält den Fehlerwert der letzten Ausführung von "eval".	Globale Spezialvariable von Perl, bei fehlerfreier Ausführung ist der Wert Null.
$Verbose	Sorgt bei "true" für eine erweiterte Ausgabe.	Wird durch "-v" beim Aufruf des Scripts gesetzt.

$Nml	Enthält den String, der jeder Ausgabe auf "StdOut" vorangestellt wird.	Dient der optischen Verbesserung der Anzeige.
$DBSGN	Enthält den String, der jeder Ausgabe von Debuginformationen auf "StdOut" vorangestellt wird.	Dient der optischen Verbesserung der Anzeige.
$DBLVL	Enthält den Wert des Debug-Levels.	Erlaubt "0", "1", oder "2"
$IpAdr	Enthält die IP-Adresse.	
$MAC	Enthält die MAC-Adresse.	
$Buffer	Puffer zur Zwischenspeicherung.	Wird beim Einlesen von Dateien verwendet.
$CmpName	Enthält den künftigen Computernamen.	
$Oktett3	Enthält die letzten 3 Oktetts der IP-Adresse.	
$Net	Enthält den Wert von Net.	
$HostMask	Enthält den Wert der HostMask.	
$Suffix	Enthält das Suffix für den späteren Computernamen.	
$CmpNameLength	Enthält den Wert für die maximale Länge des Computernamens.	Der Wert ist i. d. R. "15".
$SysprepTpl	Enthält den Wert für die Vorlagendatei zur Erstellung der *sysprep.inf*.	Z.B. "*sysprep2.inf*"
$SysprepSrchStr	Enthält den Suchstring, nach dem in $SysprepTpl gesucht wird.	Z.B. "Computername="
$Sysprep	Enthält die spätere *sysprep.inf*.	
$IniFileName	Enthält den Wert für die INI-Datei.	Z.B. "*ip2name.ini*"
$cfg	Ein Objekt vom Typ "IniFiles". Dient zum Einlesen der INI-Datei.	
$Mode	Enthält den Wert für den Modus, also "ip", "local" oder "http".	Zukünftig auch "ftp"
$FallBack	Enthält den Wert für Fallback-Modus.	true=1, false=0
$MacInputFile	Enthält den Wert zur Datei, die die MAC-Adresse enthält.	Z.B. "*input.txt*"
$MacFileSrchStr	Suchstring innerhalb von $MacInputFile.	Z.B. "Physikalische Adresse . . . . . . : " innerhalb der *input.txt*

$HTTPHost	Enthält den Wert für den http-Host.	Z.B. "*www.name.org*"
$HTTPurl	Enthält den Wert der url auf dem http-Host.	Z.B. "*/ip2name/ip2name.txt*"
$VmxFile	Enthält den Wert der *vmx*-Datei.	Z.B. "*vmpc.vmx*"; kann auch als Befehlszeilen-parameter übergeben werden.
$VmMAC	Enthält die MAC-Adresse, die per Befehlszeile übergeben worden ist.	
$VmxSearchStr1	Enthält einen Suchstring, an dessen Stelle in $VmxFile eine Ersetzung stattfinden soll.	Z.B. "uuid.location"
$VmxSearchStr2	Enthält einen Suchstring, an dessen Stelle in $VmxFile eine Ersetzung stattfinden soll.	Z.B. "uuid.bios"
$VmxSearchStr3	Enthält einen Suchstring, an dessen Stelle in $VmxFile eine Ersetzung stattfinden soll.	Z.B. "ether-net0.generatedAddress"
$VmxSearchStr4	Enthält einen Suchstring, an dessen Stelle in $VmxFile eine Ersetzung stattfinden soll.	Z.B. "displayName"

Tabelle 7.7 Übersicht der Variablen in *ip2name.pl*

Einschränkungen und Voraussetzungen

Das Script arbeitet nur mit der ersten gefundenen IP- bzw. MAC-Adresse und setzt IPv4 voraus, die Version 6 wird (noch) nicht unterstützt. Endgeräte haben meist nur einen Ethernet-Adapter und sind daher i. d .R. unkritisch. Sind bei einem Server zwei oder mehr Adapter vorhanden, dann darf zum Zeitpunkt der Ausführung des Scripts nur ein Adapter aktiv sein und eine gültige IP-Adresse besitzen. Vorgesagtes gilt natürlich auch für Endgeräte mit mehr als einem Netzwerkadapter. Die IP- bzw. MAC-Adresse des aktiven Adapters wird zur Namensbildung bzw. zum Abgleich MAC-Adresse und korrespondierendem Rechnernamen benutzt. Soll z.B. ein Server im späteren Betrieb zwei aktive Netzwerkadapter haben, dann ist die beste Vorgehensweise während des Image-Vorgangs kein Netzwerkkabel auf den zweiten Adapter zu stecken.

Das Script ist auf der Perl-Version 5.10.0 entwickelt und getestet worden. Im nächsten Kapitel sind alle Module kurz erklärt, die das Script benutzt. Module werden durch das Schlüsselwort "use" eingebunden und besitzen die Dateiendung *pm*. Prinzipiell sollte das Script auf allen Versionen >= 5.8 lauffähig sein. Wichtig ist, dass die u. g. Perl-Module von der Version unterstützt werden.

Das Script braucht Schreibrechte in dem Ordner, wo die Datei *sysprep.inf* geschrieben werden soll. Da der gesamte Prozess der Systemvorbereitung als Administrator ausgeführt wird, sollte das im produktiven Einsatz kein Problem sein.

Das Script versucht seine INI-Datei immer zuerst unter *c:\sysprep* einzulesen. Wird dort keine gefunden, wird die Datei *ip2name.ini* im aktuellen Verzeichnis gesucht. Wird auch dort keine gefunden, startet das Script mit internen Default-Werten, die denen aus der Datei *ip2name.ini.default* entsprechen.

Möglichkeiten und Risiken bei der Verwendung des Parameters **HostMask** bzw. **Net**
Die Verwendung des Parameters `Net` garantiert die Eindeutigkeit von Rechnernamen innerhalb eines Netzes. Für ein Netz der Klasse C ist der Parameter `Net = C` und intern wird somit automatisch die `HostMask` auf den Wert 111 gesetzt. Die `HostMask` kann normalerweise Werte von 111 bis 111111111 annehmen. Mit dem Wert 111 wird das letzte Oktett der IP-Adresse des Hosts bei Bedarf auf drei Stellen durch führende Nullen aufgefüllt.

IP-Adresse des Rechners	=	172.16.50.90
NetMask	=	255.255.255.0
Net	=	C
HostMask	=	111
Suffix	=	PC
Resultierender Computername	=	PC090

Tabelle 7.8 Beispiel **Net = C**

Durch die Einmaligkeit der IP-Adresse und das Auffüllen mit führenden Nullen kann im obigen Beispiel somit Eindeutigkeit bei der Bildung des Rechnernamens in einem Netz gewährleistet werden. Sind jedoch 2 Netze miteinander verbunden, ist die Eindeutigkeit der Rechnernamen gefährdet, da der Netzanteil nicht bei der Bildung der Namen berücksichtigt wird.

		Netz 1	Netz 2
IP-Adresse des Rechners	=	172.16.50.90	172.16.60.90
NetMask	=	255.255.255.0	255.255.255.0
Net	=	C	C
HostMask	=	11111	11111
Suffix	=	PC	PC
Resultierender Computername	=	PC50090	PC60090

Tabelle 7.9 Beispiel **Net = C** und **HostMask = 11111**

Durch Vergrößerung der `HostMask` kann die Eindeutigkeit im obigen Fall wiederhergestellt werden. In großen gerouteten Netzen ist der Einsatz der `HostMask` meistens unverzichtbar. Eine Vergrößerung des Wertes der `HostMask` sorgt automatisch auch für einen längeren resultierenden Computernamen.
Sollen beispielsweise 50 Terminalserver installiert werden, die IP-Adressen von 172.16.6.1 bis 172.16.6.50 besitzen, und die Namen *CTXSRV01* bis *CTXSRV50* besitzen, so kann das durch folgende Parameter erreicht werden:

IP-Adresse des Terminalservers	=	172.16.6.17
NetMask	=	255.255.0.0
Net	=	C
HostMask	=	11
HostMaskCheck	=	0
Suffix	=	CTXSRV
Resultierender Computername	=	CTXSRV17

Tabelle 7.10 Beispiel **HostMask = 11** und **HostMaskCheck = 0**

Durch Setzen des Parameters `HostMaskCheck = 0`, kann die `HostMask` kleiner als 111 gewählt werden, da das Überprüfen der Länge der `HostMask` damit abgeschaltet wird.

7.8.2 Module, Objekte, Klassen, Pakete und Methoden bei Perl

Ein Paket liegt in Form einer Datei vor, die den gleichen Namen hat, wie das Paket und die Dateinamenserweiterung *pm* besitzt. Ein Paket wird mit dem Befehl "use" eingebunden, z.B. mit "use Config::IniFiles". Damit ist ein Zugriff auf die Funktionen des Pakets möglich. Jedes Paket hat seinen eindeutigen Namensraum, der durch das Zeichen "::" als Paketseparator angezeigt wird. Für das obige Beispiel bedeutet dies, dass das Modul "IniFiles" in der Bibliothek von Perl in der Verzeichnisstruktur an der Stelle *\Config\IniFiles.pm* zu finden ist. Die Bibliothek (library) befindet sich im Unterverzeichnis *lib* im Hauptverzeichnis von Perl. Ein Modul ist ein wieder verwendbares Paket, das in einer Bibliotheksdatei abgelegt ist. Ein Paket kann aus mehreren Modulen bestehen.
Ein Paket bzw. ein Modul kann eine Klasse implementieren. Ein Objekt "$cfg" der oben genannten Klasse "IniFiles" wird beispielsweise wie folgt erzeugt:

```
$cfg = Config::IniFiles->new(-file => " ip2name.ini ");
```

Dabei wird die Methode "new" mit der Option "-file" und dem dazugehörigen Parameter "ip2name.ini" aufgerufen. Eine alternative Schreibweise lautet wie folgt:

```
$cfg = new Config::IniFiles -file => "ip2name.ini";
```

Das vorgestellte Paket dient dazu auf eine INI-Datei zuzugreifen und Variablen Werte aus dieser INI-Datei zuzuweisen. Die INI-Datei ist in verschiedene Sektionen unterteilt, die in eckigen Klammern [ ] markiert werden.
Nun kann das Objekt mithilfe der Methoden der Klasse aufgerufen werden, wobei der Rückgabewert einer Variablen zugewiesen werden kann, z.B.

```
$SysprepTpl = $cfg->val('SOURCE','SysprepTpl');
```

Hier wird der Variablen "$SysprepTpl" über die Methode "val" der Wert "c:\sysprep\sysprep2.inf" des Schlüssels "SysprepTpl" in der Sektion "SOURCE" zugewiesen. Der Abschnitt der gelesenen *ip2name.ini* schaut wie folgt aus:

```
[SOURCE]
SysprepTpl=c:\sysprep\sysprep2.inf
```

Im Programm *ip2name.pl* werden die folgenden Module benutzt:

Modulname	Benutze Funktionen und Methoden	Erläuterung
strict	*pragma*	Beschränkt unsichere Konstrukte; alle globalen Variablen müssen in 'vars' definiert sein.
Config::IniFiles	new val	Erzeugt ein neues Objekt. Wert einlesen
Sys::Info::OS	new is_unknown ip	Erzeugt ein neues Objekt. Liefert "true", wenn das Betriebssystem nicht unterstützt wird. Gibt die IP-Adresse zurück.
Data::Dumper	-	Wird von Sys::Info::OS benötigt.
String::Util	hascontent nospace	Liefert "true", wenn ein übergebener String nicht leer ist. Entfernt alle Leerstellen aus einem String.
LWP::Simple	getstore is_success	Speichert ein Dokument aus der übergebenen url in eine lokale Datei. Liefert "true", wenn der http-Status-Code "ok" ist.
vars	*pragma*	Enthält alle definierten globalen Variablen.

Tabelle 7.11 Benutzte Module und deren Funktionen/Methoden

In der vorliegenden Tabelle sind "strict" und "vars" keine Module oder Pakete, obwohl sie in Perl mit "use" deklariert sind. Diese beiden Einträge sind Compiler-Hinweise - in Perl als *pragma* bezeichnet und - weisen den Compiler an, nur lexikalisch deklarierte Variabeln zuzulassen. Diese Deklaration muss dann in "vars" erfolgen. Dagegen werden lokale Variablen mit dem Schlüsselwort "my" eingeleitet und besitzen ihren Geltungsbereich nur innerhalb eines umschließenden Blocks, einer Unterroutine oder innerhalb einer "eval"-Anweisung.

7.8.3 Beschreibung des Algorithmus

Der Kernalgorithmus besteht in der Version 0.2 des Scripts aus dem unten dargestellten Ablauf. In der Version 0.1 wurde die IP-Adresse noch aus einer Textadresse namens *input.txt* herausgefiltert, die über den Befehl `ipconfig >input.txt` gewonnen wurde. Dies ist ab der Version 0.2 nicht mehr notwendig, dort wird die IP-Adresse direkt aus dem System ausgelesen.

Nr.	Aktion	Beispiel	Codestelle
1.	Die eigene IP-Adresse bestimmen.	172.16.50.90	&GetIp
2.	Die letzten drei Oktette der IP-Adresse extrahieren und die Punkte entfernen.	16 50 90	&GetOktett
3.	Die drei Oktette bei Bedarf mit	016 050 090	&FillOktett

	führenden Nullen auf drei Stellen auffüllen.		
4.	Die HostMask bestimmen.	11111	&SetHostMask
5.	Aus dem "Suffix" und Länge der "HostMask" den Namen generieren.	Suffix=PC HostMask=11111 Name=*PC50090*	&MakeName
6.	Die Vorlagendatei *sysprep2.inf* in "$Buffer" einlesen.	*c:\sysprep\sysprep2.inf*	&GetFile
7.	In "$Buffer" nach 'Computername=' suchen und dahinter "$Name" schreiben.	Computername=*PC50090*	&ModifyBuffer
8.	Den Inhalt von "$Buffer" in sysprep.inf schreiben.	*c:\sysprep\sysprep.inf*	&WriteSysprep

Tabelle 7.12 Algorithmus *ip2name.pl* in der Version 0.2

Ab der Version 0.3 des Scripts ist es möglich, die Zuordnung des Rechnernamens über die MAC-Adresse zu vollziehen. Dies geschieht anhand einer Liste mit Wertepaaren "MAC-Adresse Rechnername". Diese Liste namens *ip2name.txt* kann lokal liegen oder alternativ zuerst über das Protokoll http heruntergeladen werden. Der Algorithmus wurde dazu in zwei Kernabläufe zerlegt, wobei die Unterscheidung anhand einer Variable "$mode" getroffen wird. Besitzt diese den Wert "ip" wird der Name nach dem Schema aus Version 0.2 gebildet. Weist sie den Wert "local" auf, wird eine lokale Liste anhand der Datei *ip2name.txt* benutzt. Bei einem Wert "http" wird diese Liste zuerst von einer url (http://...) heruntergeladen und lokal unter dem Namen *ip2name.txt* zwischengespeichert. In dieser Version wird die MAC-Adresse der Netzwerkschnittstelle benutzt. Diese wird aus einer Textdatei namens *input.txt* gewonnen. Dazu wird der Betriebssystembefehl `ipconfig –all >input.txt` benutzt. Der Ablauf der Version 0.3 sieht dann im Modus "http" bzw. "local" wie nachfolgend dargestellt aus. Dabei wird die Datei im Modus "http" vorher heruntergeladen (Schritt 3.), im Modus "local" wird sie lokal erwartet.

Nr.	Aktion	Beispiel	Codestelle
1.	Die eigene IP-Adresse bestimmen.	172.16.50.90	&GetIp
2.	Die MAC-Adresse aus der *input.txt* auslesen.	00:11:22:33:44:55	&GetMac
3.	Die Datei *ip2name.txt* in "$Buffer" einlesen, evtl. vorher herunterladen.	*c:\sysprep\ip2name.txt*	&GetFile
4.	In "$Buffer" nach MAC-Adresse suchen und Name zurückgeben.	$Name=*Pluto*	&FindMac
5.	Die Vorlagendatei *sysprep2.inf* in "$Buffer" einlesen.	*c:\sysprep\sysprep2.inf*	&GetFile

6.	In "$Buffer" nach 'Computer-name=' suchen und dahinter "$Name" schreiben.	Computername=*Pluto*	&ModifyBuffer
7.	Den Inhalt von "$Buffer" in *sysprep.inf* schreiben.	*c:\sysprep\sysprep.inf*	&WriteSysprep

Tabelle 7.13 Algorithmus *ip2name.pl* in der Version 0.3

Die Bestimmung der IP-Adresse ist dort streng genommen nicht notwendig; sie kann aber an dieser Stelle als Abbruchkriterium genutzt werden. Kann keine IP-Adresse bzw. nur die Adresse 'localhost' mit der IP-Adresse 127.0.0.1 bestimmt werden, kann von einem Fehler ausgegangen werden.

In der Version 0.4 wurden zum einen Anpassungen vorgenommen, um den Einsatz in einer VMWare-Umgebung zu unterstützen. Dazu wird das Script mit dem Parameter "-vm" gestartet. Zum anderen wurde ein Fallback-Modus ins Leben gerufen, der es im Modus "http" auf Wunsch ermöglicht, bei einem Fehler beim Herunterladen oder Einlesen der Datei *ip2name.txt* in den Modus "ip" zurückzuwechseln.

Wird das Script mit dem Befehlszeilenschalter "-vm" aufgerufen, verhält es sich völlig anders als im "normalen" Betrieb. Es wäre auch denkbar gewesen, für diesen Fall ein separates Script zu schreiben. Nach reiflicher Überlegung verdrängte jedoch die Möglichkeit ohne viel Aufwand fertige Routinen zu nutzen, die Verwendung eines zweiten Scripts in den Hintergrund. Der genaue Ablauf des Scripts wird an dieser Stelle zugunsten der Lesbarkeit noch nicht erläutert. Die notwendigen Grundlagen zum Verstehen der Abläufe werden erst in den folgenden Kapiteln "Installation der Testanlage" und "Automatische Softwareinstallation der virtuellen Clients" gelegt. Im Diagramm des Kapitels "Ablaufdiagramm des Algorithmus" ist der Ablauf des Scripts mit dem Schalter "-vm" zur besseren Unterscheidung gestrichelt gekennzeichnet.

Die möglichen Parameter für das Script lauten "-makeini", "-vm", "-v" und "-help". Dabei ist immer nur ein Parameter zur gleichen Zeit gestattet. Der Parameter "-v" ist vielleicht von anderen Programmen geläufig und bietet hier die erweiterte Ausgabe von Informationen während der Laufzeit des Scripts. Der Parameter "-v" steht dabei für die Abkürzung von "verbose"[30].

Auch der Parameter "-help" lässt vermuten, dass er eine kurze Hilfestellung zur Verwendung des Programms ausgibt, die dann wie folgt aussieht:

```
ip2name.pl [-h|-makeini|-test|-v|-vm]
        -h:       display this help
        -makeini: generates a default ini-file named
                                        'c:\sysprep\ip2name.ini.default'
        -v:       verbose, display a little bit more...
        -vm:      --MAC-Adress [--VMwareVmxFile]
        --MAC-Adress:    in format '11:22:33:44:55:66'
        --VMwareVmxFile: complete path to the .vmx-file; if missing the
                          value will be read from the ini-file

example: ip2name.pl -vm 11:22:33:44:55:66 d:\vmware\mymachine.vmx

Prerequisites: (a) installed perl ;-)
               (b) perl-modules:  Config::IniFiles
                                  Sys::Info::OS
                                  Data::Dumper
```

[30] verbose; (engl.), zu deutsch wortreich

```
                     String::Util
                     LWP::Simple
(c) a ini-file named 'c:\sysprep\ip2name.ini' or 'ip2name.ini'
     in the same directory
     recommended: 'c:\sysprep\ip2name.ini'
(d) a template file named 'sysprep2.inf' for building a
     new sysprep.inf
(e) if not runnin' in 'ip'-mode, a file 'input.txt', where
     to find the own MAC-Adress of the active interface,
     e.g. generated with 'ipconfig -all >input.txt' before
     calling ip2name.pl
(f) suffient rights in the directory to write 'sysprep.inf'
(g) in 'virtual Mode', suffient rights in the directory to
      change the vmx-File in the given VMWare-directory
```

In der ersten Zeile taucht auch noch der Hinweis zu einem Parameter namens "-test" auf. Dieser war bis zur Version 0.2 Bestandteil des Scripts und sollte in erster Linie einen ausführlichen Test des Scripts bewirken, der auch optisch ansprechend präsentiert werden sollte. Dieser wurde aufgrund von Zeitmangel und zugunsten einer ausführlichen Debug-Ausgabe in der Version 0.4 entfernt. Der Debug-Level wird in der *ip2name.ini* gesteuert und in der Default-Einstellung erfolgt keine Ausgabe. Der Test-Parameter wird aber ab der Version 0.5 wieder unterstützt werden.

Der Aufruf des Scripts mit dem Befehlszeilenparameter "-makeini" erzeugt im aktuellen Verzeichnis eine INI-Datei namens *ip2name.ini.default*. Die Werte innerhalb dieser Datei spiegeln zugleich die internen Default-Einstellungen wider. Diese werden bei fehlenden Parametern automatisch benutzt. Die INI-Datei kann dann entsprechend den Anforderungen editiert und für das Script genutzt werden, indem sie in *ip2name.ini* umbenannt wird.

7.8.4 Ablaufdiagramm des Algorithmus

Der Programmablaufplan der Version 0.4 ist in der nachfolgenden Zeichnung dargestellt. Er beinhaltet alle optionalen Parameter und stellt in einzelnen Schritten den Gesamtablauf dar. Dabei repräsentiert ein Schritt i. d. R. die gerade aktive Unterroutine, die mit einem "&" gekennzeichnet ist. Die Bezeichnung "CL-Options" steht für CommandLine-Options, also in der deutschen Übersetzung die Befehlszeilenparameter. Die rautenförmigen Zeichen symbolisieren Abfragen und Vergleiche, die mit "yes" oder "no" ausgewertet werden. Wird ein ungültiger Befehlszeilenparameter übergeben, endet das Programm mit dem Hinweis auf den Parameter "-h" zur Anzeige der Hilfefunktion.

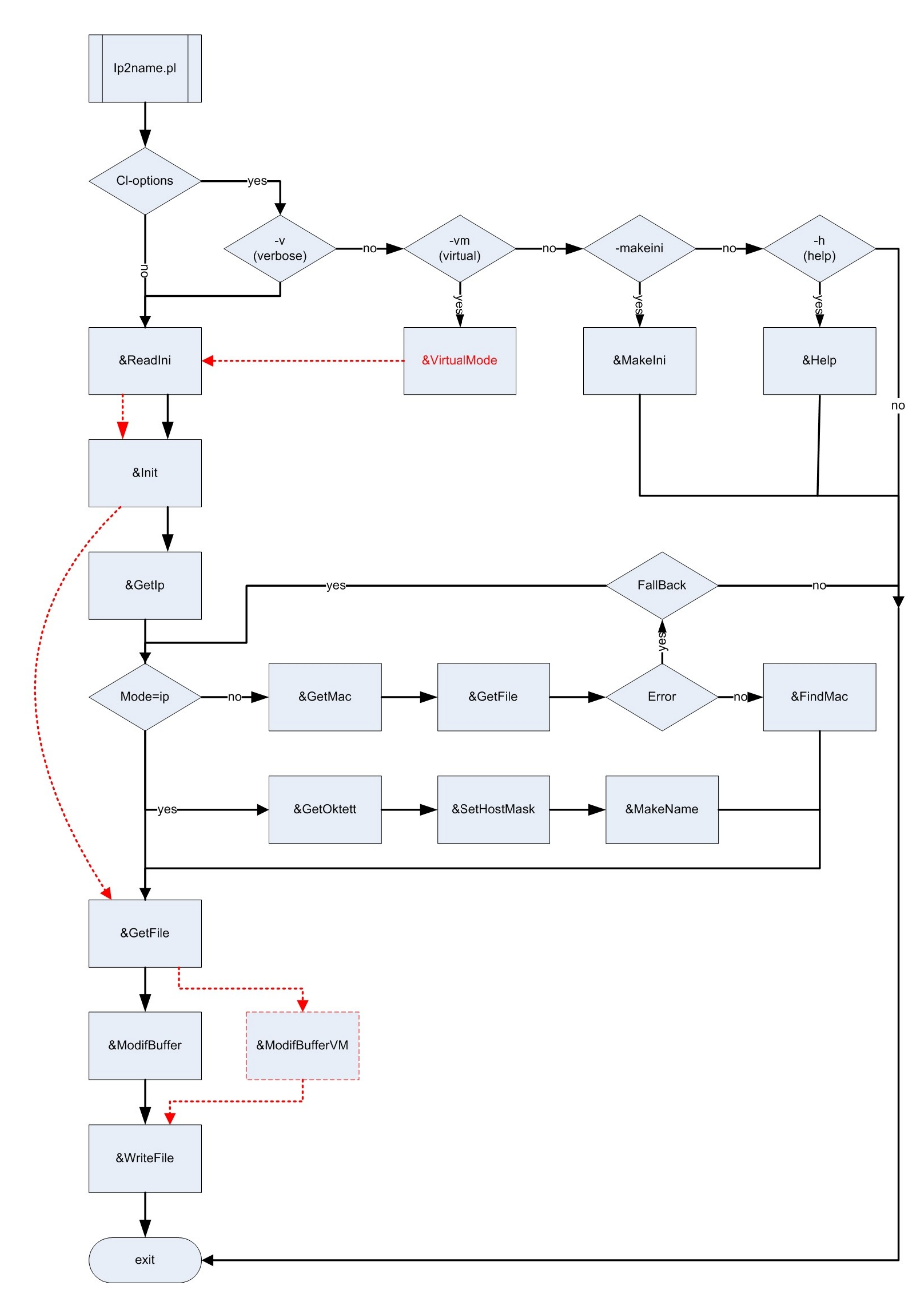


Abbildung 7.29 Programmablaufplan ip2name.pl Version 0.4

Wie schon weiter oben ausgeführt, ist der Verlauf, der bei der Benutzung des Parameters "-vm" ausgeführt wird, optisch durch gestrichelte Linienführung kenntlich gemacht.

7.8.5 Programmierung/Codierung

Anhand eines kurzen Abrisses der Hauptroutine "main" soll der prinzipielle Ablauf des Scripts erläutert werden. Das Paket "main" ist in jedem Perl-Script aktiv, muss also nicht explizit mit "use" definiert werden und verweist auf den Pfad *\lib* innerhalb. Dieses Paket stellt die Grundfunktionen von Perl zur Verfügung. Andere benutzte Paket werden mit dem Schlüsselwort "use" versehen und am besten am Anfang des Scripts platziert.
Das Script beginnt nach einigen informativen und kosmetischen Zeilen mit der Auswertung der übergebenen Befehlszeilenparameter. Wurden keine übergeben, wird die while-Schleife nicht betreten.

```
while ($_ = $ARGV[0])
      {my $VmMAC = $ARGV[1];
      $VmxFl = $ARGV[2];
      shift;
      last if /^^--$/;
      if (/^-h(.*)/) {&Help}
      elsif (/^-makeini$/) {&MakeIni}
      elsif (/^-v$/) {$Verbose = "true"}
      elsif (/^-vm$/) {&VirtualMode ($VmMAC, $VmxFl)}
      else {print (":-( ignoring unknown parameter $_, use -? to show options.
:-)\n")};
```

Befehlszeilenparameter werden im Array `@ARGV` gespeichert und durch reguläre Ausdrücke `/^.../` mit erlaubten Werten verglichen und dann die zugehörige Unterroutine `&...` gestartet.
Ohne Parameter fährt das Script mit dem Hauptteil fort.

```
&ReadIni ("");
&Init ("");
```

Das Unterprogramm `ReadIni` versucht die INI-Datei unter *c:\sysprep\ip2name.ini* oder im aktuellen Verzeichnis zu finden und zu laden. Schlägt dies fehl, wird die Variable `IniFileName` auf Null gesetzt und in der Folge in der Unterroutine `Init` die internen Default-Werte geladen. Konnte die INI-Datei *ip2name.ini* erfolgreich gefunden werden, können die dort definierten Werte in die Variablen geladen werden, z.B.

```
    $HostMaskCheck = $cfg->val('TARGET','HostMaskCheck');
    if (!hascontent $HostMaskCheck)
      {$HostMaskCheck = 1;
      print($message, "HostMaskCheck=", $HostMaskCheck, "\n")
      };
```

Kann ein Wert in der INI-Datei nicht gefunden werden,

```
 if (!hascontent $HostMaskCheck)
```

wird er mit einem Default-Wert belegt, hier: `$HostMaskCheck = 1;`
Nach der Initialisierung wird die IP-Adresse des Hosts in der Funktion `GetIp` bestimmt, in der Variable `IpAdr` gespeichert und im Anschluss mit `print` ausgegeben.

```
print ($Nml, "IP-Adress='", $IpAdr = &GetIp, "'\n");
```

In Abhängigkeit vom Arbeitsmodus des Scripts - gespeichert in der Variable `mode` - wird der Computername bestimmt. Ist der Modus nicht der Default-Modus "ip", wird die MAC-Adresse für die Zuordnung des Rechnernamens benutzt. Diese wird aus der Datei *input.txt* gewonnen. Innerhalb dieser Datei wird nach dem String `MacFileSrchStr` gesucht, dabei wird in der Variable `MACSize` die Länge der MAC-Adresse vom ersten bis zum letzten Zeichen angegeben. Die Länge beinhaltet auch mögliche Trennzeichen, also beträgt die Länge des MAC-Adressen-Strings in diesem Fall "11:22:33:44:55:66" 17 Zeichen.

Anschließend wird die Datei mit der Zuordnung MAC-Adresse und Rechnername über die Funktion `GetFile` eingelesen.

```
if ($Mode ne "ip")
   {print ($Nml, "MAC-Adress='", $MAC = &GetMac($MacFileSrchStr, , $BufferSize,
$MACSize), "'\n");
    eval { $Buffer = &GetFile($LocalInputFile, $BufferSize, $Mode)};
         if ($@ ne "")
            {print ("\nFailure while reading buffer, $@\n");
             if ($FallbackMode == 1)
                {$Mode = "ip";
                 print ("=> working in fallback-mode, changed mode='",
$Mode,"'\n");
                }
             else {exit};
            }
         else {$CmpName = &FindMac($Buffer, $MAC);};
   };
```

Schlägt dieses Einlesen fehl, wird eine Fehlermeldung ausgegeben. Ist der Fallback-Modus aktiv (`$FallbackMode == 1`), dann wird der Wert der Variable `mode` auf den Wert "ip" gesetzt. War das Einlesen erfolgreich, befindet sich der Inhalt der eingelesenen Datei in der Variable `Buffer`. Im Anschluss wird über die Funktion `FindMac` die MAC-Adresse des Hosts im Inhalt von `Buffer` gesucht und der zugehörige Computername in `CmpName` zurückgegeben. Wird die MAC-Adresse nicht gefunden, wird ein leerer Computername zurückgegeben.

Hat die Variable `mode` nun den Wert "ip" wird die Funktion `GetOktett3` aufgerufen, die einen 9-stelligen String zurückliefert, der aus den letzten drei Oktetten der IP-Adresse besteht, die bei Bedarf mit führenden Nullen auf je drei Stellen ergänzt worden sind.

```
  if ($Mode eq "ip")
     {#get the last 3 oktetts, will be filled to 3 each
      $Oktett3 = &GetOktett;
      #set the length of the hostmask, will be from 3...9
      #111=3, 111111111=9
      $HostMask = &SetHostMask($HostMask);
      #build computername
      $CmpName = &MakeName($Suffix, $Oktett3, $HostMask, $CmpNameLength);
     };
if ($CmpName eq "") {warn ("!!! Computername='$CmpName' ist empty !!!\n")}
```

Im weiteren Ablauf wird die HostMask gesetzt und der Computername durch die Funktion `MakeName` gebildet. Diese erwartet die Parameter `Suffix`, `Oktett3`, `HostMask` und `CmpNameLength`. Ist der Computername leer, wird eine Warnung ausgegeben.

Nun muss noch die Vorlagendatei *sysprep2.inf* in die Variable `Buffer` eingelesen werden.

```
eval { $Buffer = &GetFile($SysprepTpl, $BufferSize)};
if ($@ ne "")
   {print ("\nFailure while reading buffer, $@\n");
    exit;}
```

Tritt dabei ein Fehler auf, wird das Script mit `exit` verlassen. Im Folgenden wird der Inhalt von `Buffer` geprüft. Ist er leer, wird das Script mit `exit` verlassen, ansonsten wird er über `ModifyBuffer` geändert. Dabei wird in `Buffer` nach `SysprepSrchStr` gesucht und hinter diese Stelle wird der Inhalt von `CmpName` geschrieben.

```
if (hascontent $Buffer)
   {$Buffer = &ModifyBuffer($Buffer, $SysprepSrchStr, $CmpName);
    if ($Buffer ne "false") {if ($DBLVL == 0, $Verbose) {print ("=>
successfull\n")};}
     else {print ("=> Error in module \$ModifyBuffer!\n");}
    }
else
```

```
    {print ("Error: Buffer is empty, maybe $SysprepTpl to?!?\n Exiting...");
     exit};
```

Am Schluss wird die Datei *sysprep.inf* erzeugt. Dies erledigt die Unterroutine `WriteFile`, die als Parameter den Dateinamen inkl. Pfad (`Sysprep`), den Inhalt der Datei (`Buffer`) und ein optionales Addendum (`Add`), das ans Ende der Datei geschrieben wird, erwartet.

```
my $Add = ("[$Script]\n"."$CmpName=written by $Script $Version
(time='".&GetTimeStamp."')\n");
eval {&WriteFile($Sysprep, $Buffer, $Add)};
if ($@ ne "")
  {print ("\nFailure while writing $Sysprep, $@\n");
   print ("!!! $0 not finished sucessfully!!!\n")}
else
   {print ("==> $0 finished!\n");}
exit;
```

Die Hauptroutine beendet das Script *ip2name.pl* mit `exit`.

7.8.6 Test

Die Batchdatei *ip2name.cmd* kann im Testmodus arbeiten, indem im gleichen Verzeichnis eine Datei namens *testmode.yes* angelegt wird. Testmodus bedeutet an dieser Stelle, dass alles abgearbeitet wird, lediglich der Aufruf der *sysprep.exe* und das Rücksetzen des Administratorkennworts erfolgt nicht.
Vergleichbares gibt es für *ip2name.pl* nicht. Das Programm erzeugt auch "nur" eine neue Datei *sysprep.inf*, die mit jedem beliebigen Editor geöffnet werden kann. Eine Art von Testmodus ist insofern realisiert, als dass es möglich ist, über die zugehörige INI-Datei *ip2name.ini* sämtliche Parameter des Programms zu verändern. Damit können dann vorab alle zu erwartenden Test-Szenarien durch Veränderung der Parameter evaluiert werden.
Die INI-Datei wird zuerst im Verzeichnis *c:\sysprep* gesucht. Wird sie dort nicht gefunden, versucht das Programm die Datei im aktuellen Verzeichnis zu lokalisieren. Auf der beiliegenden CD befindet sich das Verzeichnis *\ip2name*, welches den Perl-Interpreter, das Perl-Script und einige Beispieldateien beinhaltet. Dieses Verzeichnis kann auf den lokalen Rechner kopiert werden. Zum Ausführen des Scripts sind Schreibrechte im entsprechenden Verzeichnis notwendig. Es ist ebenfalls ratsam, den Schreibschutz von den Dateien zu entfernen, der nach dem Kopieren von der CD auf den Dateien liegt.
Auf der CD befindet sich im Unterverzeichnis *\ip2name\proton* ein kostenfreier Editor, der durch den Aufruf von *proton.exe* gestartet werden kann. Dieser beherrscht Syntax-highlighting und ermöglicht so einen besseren Überblick über das Programm. Mit diesem Editor wurde das Programm auch bearbeitet.
In der kommenden Version 0.5 wird ein Parameter "test" eingeführt werden, der es ermöglicht, eine ausführliche Diagnose am Bildschirm auszugeben und diese auf Wunsch in einer Reportdatei zusammenzufassen. Der Beginn der Entwicklung dieses Parameters war schon in der Version 0.3 enthalten, aber aufgrund von Zeitproblemen wurde die Weiterentwicklung zurückgestellt.

7.9 Manuelle Tätigkeiten

Tätigkeiten, die nicht ohne weiteres vollständig automatisiert werden können, sind

- das Auspacken der Geräte,
- die Inventarisierung der Geräte,

- das Beschriften der Geräte,
- das Erstellen der Reservierung im DHCP,
- das Aufstellen der Geräte beim Benutzer bzw. im Serverraum.

Spielen monetäre Gründen keine entscheidende Rolle, ist es durchaus denkbar, die oben genannten Tätigkeiten zu automatisieren bzw. durch externe Dienstleister erledigen zu lassen. Im Falle einer Neubeschaffung von Hardware, könnte man den Ablauf so gestalten, dass man dem Dienstleister, der die Hardware liefert, eine Liste der Rechnernamen und der Benutzer bzw. Büros zukommen lässt, wo diese Geräte aufgestellt werden sollen. Dieser beschriftet die Geräte entsprechend den Vorgaben mit Rechnername und MAC-Adresse und gibt in elektronischer Form eine Liste mit Rechnername und zugehöriger Seriennummer und MAC-Adresse und dem Bestimmungsort bzw. dem Benutzer zurück. Diese Liste könnte dann mittels eines Scripts sowohl in die Inventardatenbank eingelesen als auch zur Durchführung der DHCP-Reservierung genutzt werden. Anschließend stellt der Dienstleister die Geräte am Bestimmungsort auf, startet die BIOS-Konfiguration über PXE und darauf die Installation des Images.

Soll der Rechner beispielsweise einem Benutzer zur Verfügung gestellt werden, der erweiterte Rechte zur täglichen Arbeit braucht, ist es notwendig, den Computer-Account innerhalb des Active Directory in eine andere OU zu verschieben. Dies kann auch bei einem virtuellen Terminalserver der Fall sein, der zum Test bestimmter Einstellungen, innerhalb des AD verschoben werden muss. In beiden Fällen müssen diese Änderungen von Hand durchgeführt werden.

Auch eine Anpassung der Beschreibung des Computerobjektes im AD sowie der Beschreibung am Rechner selbst müssen derzeit manuell vorgenommen werden. Die Beschreibung des Objektes im Active Directory hat informativen Charakter und könnte aus der Inventardatenbank ausgelesen werden, wie z.B. "Referat 1.10 Hr. Mustermann". Die gleiche Beschreibung könnte bei Rechnern selbst eingetragen werden und erscheint dann, wenn man die Netzwerkumgebung durchsucht, als informativer Hinweis auf den Benutzer des Rechners. Dazu wird unter "Arbeitsplatz/Eigenschaften/Computer" im Feld "Computerbeschreibung" die vorgenannte Information eingetragen. Beide Informationen könnten künftig ebenfalls gescriptet werden.

Vielfach werden bei Servern die IP-Adressen manuell festgelegt und DHCP deaktiviert. Prinzipiell ist dies nicht notwendig, jedoch erfordern ggf. spezielle Einstellungen der Server diesen Schritt. Ebenso muss eine vorhandene zweite Netzwerkkarte manuell eingerichtet werden oder zumindest nach dem Imagevorgang manuell aufgesteckt werden, da zurzeit während des Image-Prozesses nur eine Netzwerkkarte aktive Verbindung zum Netzwerk haben darf.

Selbstverständlich sind Erstellung, Pflege und Test sowie die Dokumentation der Images manuelle Aktivitäten.

7.10 Anpassung des Images an andere Hardware

Um dem schnellen Wandel, vor allen Dingen im Rechnerbereich, gerecht zu werden, ist es unvermeidlich das Image auf neue Hardware zu verteilen. Im Idealfall sind dort keine oder kaum Anpassungen vorzunehmen, wenn die Hardware durch bereits installierte Treiber vollkommen abgedeckt wird. Im Normalfall sind aber Anpassungen unvermeidlich.
Der allgemeine Fall und der dazugehörige Ablauf werden wie folgt festgelegt:

1. Auf den Basis-PC wird das aktuelle Basisimage aufgespielt.
2. Die Datei *sysprep.inf* wird angepasst, indem die Sektion "MassStorageDevice" gefüllt wird und ggf. werden zusätzliche Massenspeichertreiber vorbereitet.
3. Ein Image namens *PCnameVersion_driver.gho* wird erstellt.
4. Das vorgenannte Image wird auf die neue Zielplattform aufgespielt.
5. Auf der neuen Hardware werden ggf. zusätzliche Treiber nachinstalliert.
6. Ein weiteres Image namens *PCnameInkrementierteVersion_factory.gho* wird erstellt; die Versionsnummer wird dabei inkrementiert.
7. Das vorgenannte Image wird auf die originäre Hardwareplattform aufgespielt.
8. Aus dem "Factory"-Modus wird wieder in den normalen Ausgangsmodus gewechselt.
9. Es wird ein neues Basisimage mit inkrementierter Versionsnummer erstellt.

In der nachfolgenden Zeichnung wird der obige Ablauf verdeutlicht. Sollte der Transfer erfolgreich verlaufen sein, können die beiden während des Ablaufs erstellten Images "Driver" und "Factory" gelöscht werden. Sie dienen nur dazu Zwischenstände festzuhalten, zu denen im Fehlerfall schnell zurückgekehrt werden kann. Als Installationsquelle wird immer die Freigabe "Installationmanager" auf dem Fileserver gewählt. An dieser Stelle liegen die Softwareinstallationen, die auch für die Terminalserver genutzt werden. Damit ist gewährleistet, dass immer äquivalente Installationen auf Terminalserver und Endgerät vorgenommen werden.

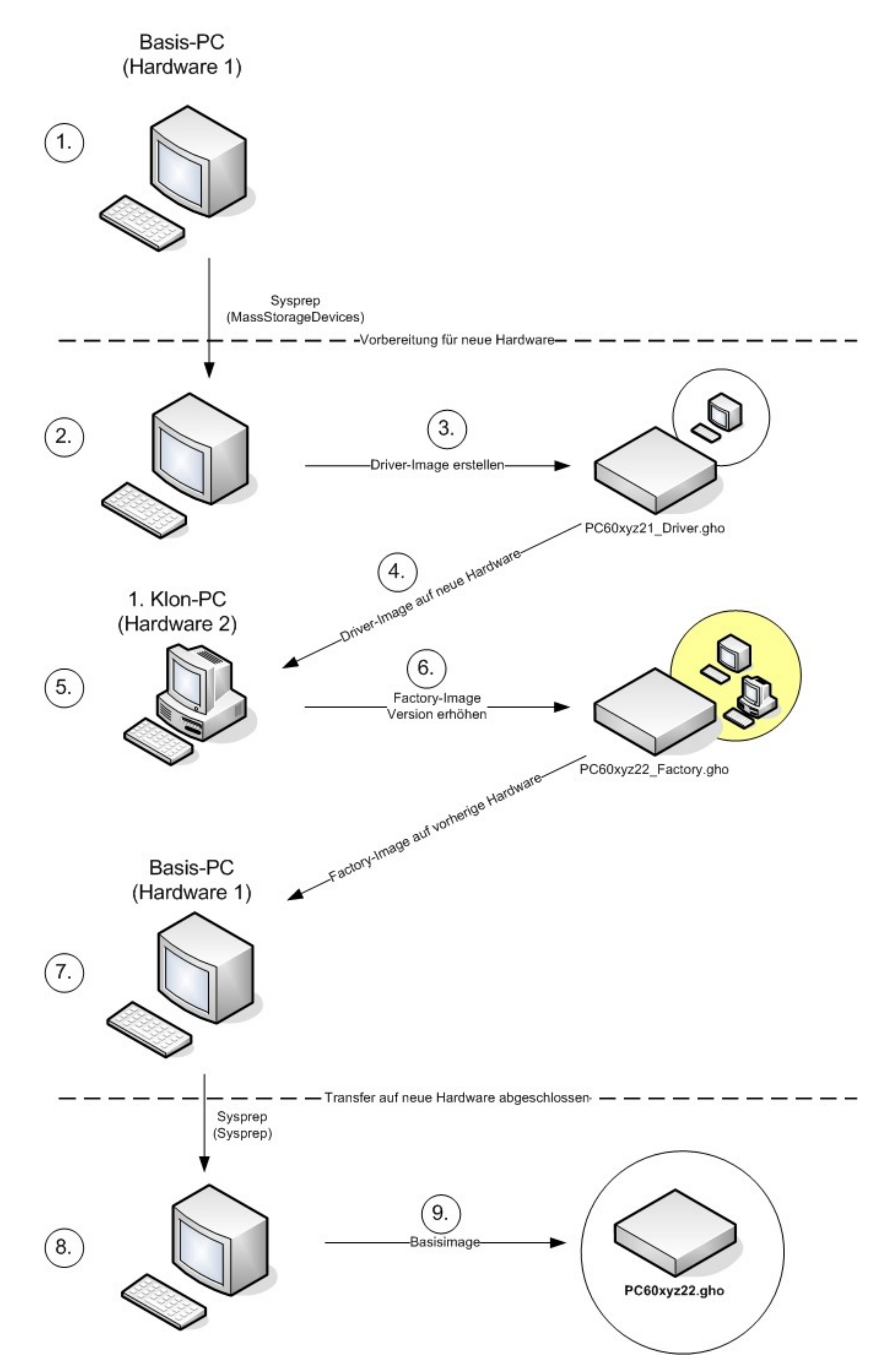


Abbildung 7.30 Transfer eines Images auf neue Hardware

Alle Installationen werden immer mit dem lokalen Account Administrator durchgeführt. Der Beginn der Erweiterung des Images ist immer das Zurückspielen des aktuellen Basisimage. Nur so ist eine konsistente Pflege und Dokumentation des Images möglich. Im folgenden Beispiel ist die Entwicklung des Images in der Version 22 auf einem FSC Esprimo E 5700 dokumentiert. In diesem Schritt wird das Image auf die neue Hardwareplattform FSC Esprimo E 8310 portiert. Es wurde bereits festgestellt, dass das vorhandene Basisimage in der Version 21 nicht direkt auf den E 8310 aufgespielt werden kann, da ein passender Mas-

senspeichertreiber für dieses Modell im Image fehlt. Der Grafiktreiber ist identisch zu dem Modell E 5915, welches in der Version 21 hinzugefügt wurde. In der Datei *sysprep.inf* beginnen Kommentarzeilen mit den Zeichen ";".
In der Dokumentation wird jeder Schritt stichwortartig und reproduzierbar mit Datum festgehalten.

06.12.2007:

- Image *PC60xyz21.gho* auf E 5700 zurückgespielt
- Ordner C:\Install\Driver_E8310 erstellt u. folgende Treiber in das Verzeichnis kopiert
  - *Audio\FSC_Realtek_High_Definition_Audio_5_10_0_5436__1015183.zip*
  - *Cardbus\FSC_O2_Micro_Flash_MemoryCard_2_16__1015318.zip*
  - *Cardbus\FSC_O2_Micro_OZ711MP1_MS1_SmartCard_Controller_and_O2M_3_0_1.zip*
  - *Chipset\FSC_Intel_Mobile_965GM___PM_Express_Chipset_8_2_0_1014__1010864.zip*
  - *Grafik\FSC_Mobile_Intel__R__945___965_Express_Chipset_family_6_14_10_4785__1011065.zip*
  - *LAN\FSC_Marvell_Yukon_Family_Driver_8_61_2_3__1010429.zip*
  - *SATA\Intel Storage Manager-7.8.0.1012*
  - *System Device\FSC_Fujitsu_FUJ02B1_1_21_0_0__1000500.zip*
  - *System Device\FSC_Fujitsu_FUJ02E3_1_0_0_0__1002301.zip*
  - *Touchpad\FSC_Synaptics_PS_2_Port_Touchpad_8_3_7_0__1006607.ZIP*
- 32-Bit-SATA-Treiber nach *c:\PnPDrvrs\SATA\Intel_ICH8M_7.8.0.1012\32bit* entpackt; Beachte: Im BIOS muss der AHCI-Modus auf enabled gesetzt werden!
- Datei *C:\Sysprep\sysprep.inf* geändert:

```
[Sysprep]
BuildMassStorageSection = Yes
```

- Befehl `C:\Sysprep\sysprep.exe –bmsd` ausgeführt
- Datei *C:\Sysprep\sysprep.inf* geändert:

```
;[Sysprep]
;BuildMassStorageSection = Yes

[Unattended]
OEMPnPDriversPath =
"PnPDrvrs\E5915\Graphic\Graphics;PnPDrvrs\VMware\Graphic\;PnPDrvrs\SATA\Intel_IC
H8M_7.8.0.1012\32bit\"

[BenutzerData]
ComputerName=PC60xyz21_Driver

[Identification]
    JoinWorkgroup=ARBEITSGRUPPE

[GuiUnattended]
    ;AutoLogon=Yes
    ;AutoLogonCount=1

;[GuiRunOnce]
    ;Command0=C:\Install\Sysprep\1login.cmd

[SysprepMassStorage]
;SATA 32bit fuer Lifebook E8310
iaStor_ICH8M = "Intel(R) 82801HEM SATA RAID Controller (Mobile ICH8M-E)"
iaAHCI_ICH8M = "Intel(R) 82801HEM/HBM SATA AHCI Controller (Mobile ICH8M-E/M)"
iaStor_ICH8R = "Intel(R) 82801HR/HH/HO SATA RAID Controller (Desktop ICH8R)"
iaAHCI_ICH8R = "Intel(R) 82801HR/HH/HO SATA AHCI Controller (Desktop ICH8R)"
```

```
iaStor_ESB2 = "Intel(R) 631xESB/632xESB SATA RAID Controller (Server/Workstation
ESB2)"
iaAHCI_ESB2 = "Intel(R) 631xESB/632xESB SATA AHCI Controller (Server/Workstation
ESB2)"
iaStor_ICH7MDH = "Intel(R) 82801GHM SATA RAID Controller (Mobile ICH7MDH)"
iaStor_ICH7DH = "Intel(R) 82801GR/GH SATA RAID Controller (Desktop ICH7R/DH)"
iaAHCI_ICH7R = "Intel(R) 82801GR/GH SATA AHCI Controller (Desktop ICH7R/DH)"
iaAHCI_ICH7M = "Intel(R) 82801GBM SATA AHCI Controller (Mobile  ICH7M/DH)"
iaStor_ICH6R = "Intel(R) 82801FR SATA RAID Controller (Desktop ICH6R)"
iaAHCI_ICH6R = "Intel(R) 82801FR SATA AHCI Controller (Desktop ICH6R)"
iaAHCI_ICH6M = "Intel(R) 82801FBM SATA AHCI Controller (Mobile  ICH6M)"
PCI\VEN_8086&DEV_282A&CC_0104="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaS
tor.inf", "Intel(R) 82801HEM SATA RAID Controller (Mobile ICH8M-E)"

PCI\VEN_8086&DEV_2829&CC_0106="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaa
hci.inf", "Intel(R) 82801HEM/HBM SATA AHCI Controller (Mobile ICH8M-E/M)"
PCI\VEN_8086&DEV_2822&CC_0104="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaS
tor.inf", "Intel(R) 82801HR/HH/HO SATA RAID Controller (Desktop ICH8R)"
PCI\VEN_8086&DEV_2821&CC_0106="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaa
hci.inf", "Intel(R) 82801HR/HH/HO SATA AHCI Controller (Desktop ICH8R)"
PCI\VEN_8086&DEV_2682&CC_0104="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaS
tor.inf",  "Intel(R)  631xESB/632xESB  SATA  RAID  Controller  (Server/Workstation
ESB2)"
PCI\VEN_8086&DEV_2681&CC_0106="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaa
hci.inf",  "Intel(R)  631xESB/632xESB  SATA  AHCI  Controller  (Server/Workstation
ESB2)"
PCI\VEN_8086&DEV_27C3&CC_0104="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaS
tor.inf", "Intel(R) 82801GHM SATA RAID Controller (Mobile ICH7MDH)"
PCI\VEN_8086&DEV_27C6&CC_0104="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaS
tor.inf", "Intel(R) 82801GR/GH SATA AHCI Controller (Desktop ICH7R/DH)"
PCI\VEN_8086&DEV_27C1&CC_0106="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaa
hci.inf", "Intel(R) 82801GR/GH SATA AHCI Controller (Desktop ICH7R/DH)"
PCI\VEN_8086&DEV_27C5&CC_0106="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaa
hci.inf", "Intel(R) 82801GBM SATA AHCI Controller (Mobile  ICH7M/DH)"
PCI\VEN_8086&DEV_2652&CC_0104="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaS
tor.inf", "Intel(R) 82801FR SATA RAID Controller (Desktop ICH6R)"
PCI\VEN_8086&DEV_2652&CC_0106="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaa
hci.inf", "Intel(R) 82801FR SATA AHCI Controller (Desktop ICH6R)"
PCI\VEN_8086&DEV_2653&CC_0106="C:\PnPDrvrs\SATA\Intel_ICH8M_7.0.0.1020\32bit\iaa
hci.inf", "Intel(R) 82801FBM SATA AHCI Controller (Mobile  ICH6M)"

[HardwareIds.scsi.iaStor_ICH8M]
id = "PCI\VEN_8086&DEV_282A&CC_0104","iaStor"
[HardwareIds.scsi.iaAHCI_ICH8M]
id = "PCI\VEN_8086&DEV_2829&CC_0106","iaStor"
[HardwareIds.scsi.iaStor_ICH8R]
id = "PCI\VEN_8086&DEV_2822&CC_0104","iaStor"
[HardwareIds.scsi.iaAHCI_ICH8R]
id = "PCI\VEN_8086&DEV_2821&CC_0106","iaStor"
[HardwareIds.scsi.iaStor_ESB2]
id = "PCI\VEN_8086&DEV_2682&CC_0104","iaStor"
[HardwareIds.scsi.iaAHCI_ESB2]
id = "PCI\VEN_8086&DEV_2681&CC_0106","iaStor"
[HardwareIds.scsi.iaStor_ICH7DH]
id = "PCI\VEN_8086&DEV_27C3&CC_0104","iaStor"
[HardwareIds.scsi.iaStor_ICH7MDH]
id = "PCI\VEN_8086&DEV_27C6&CC_0104","iaStor"
[HardwareIds.scsi.iaAHCI_ICH7R]
id = "PCI\VEN_8086&DEV_27C1&CC_0106","iaStor"
[HardwareIds.scsi.iaAHCI_ICH7M]
id = "PCI\VEN_8086&DEV_27C5&CC_0106","iaStor"
[HardwareIds.scsi.iaStor_ICH6R]
id = "PCI\VEN_8086&DEV_2652&CC_0104","iaStor"
[HardwareIds.scsi.iaAHCI_ICH6R]
id = "PCI\VEN_8086&DEV_2652&CC_0106","iaStor"
[HardwareIds.scsi.iaAHCI_ICH6M]
id = "PCI\VEN_8086&DEV_2653&CC_0106","iaStor"
;Windows default
```

- Befehl `C:\Sysprep\sysprep.exe -factory –quiet` ausgeführt
- Image *PC60xyz21_Driver.gho* erstellt
- Image *PC60xyz21_Driver.gho* auf E 8310 zurückgespielt
- Hardwareerkennung und notwendige Treiber aus Ordner C:\Install\Driver_E8310 installiert und ggf. Neustart
  - *Cardbus\FSC_O2_Micro_Flash_MemoryCard_2_16__1015318.zip*
  - *Cardbus\FSC_O2_Micro_OZ711MP1_MS1_SmartCard_Controller_and_O2M_3_0_1.zip*
  - *LAN\FSC_Marvell_Yukon_Family_Driver_8_61_2_3__1010429.zip*
  - *System Device\FSC_Fujitsu_FUJ02B1_1_21_0_0__1000500.zip*
  - *System Device\FSC_Fujitsu_FUJ02E3_1_0_0_0__1002301.zip*
  - *Touchpad\FSC_Synaptics_PS_2_Port_Touchpad_8_3_7_0__1006607.ZIP*
- Image *PC60xyz22_Factory.gho* erstellt
- Image *PC60xyz22_Factory.gho* auf E 5700 zurückgespielt
- Hardwareerkennung abgewartet und Rechner neu gestartet
- Datei *C:\Sysprep\sysprep.inf* geändert:

```
[BenutzerData]
ComputerName=PC60xyz
```

- Befehl `C:\Sysprep\sysprep.exe -mini -activated -reseal -quiet -nosidgen -noreboot` ausgeführt
- Datei *C:\Sysprep\sysprep.inf* geändert:

```
[GuiUnattended]
    OEMDuplicatorstring=PC60xyz-Baseimage-Sysprep-06122007
```

- Hintergrundbild Desktop_MIG_1024x768_V22.jpg auf Version 22 vom 06.12.2007 geändert
- Umgebungsvariable angepasst: ImageFile=V.22-06.12.07
- Image PC60xyz22.gho erstellt

8 Virtualisierung

Die Virtualisierung ist eine Methode die Ressourcen einer Hardwareplattform aufzuteilen und verschiedenen Betriebssystemen zur selben Zeit zur Verfügung zu stellen. Völlig isoliert von dem eigentlichen Host-Betriebssystem und damit in absolut sicherer Umgebung eignen sich diese virtuellen Systeme vor allem für autarke Testzwecke. So können Anwendungen und neue Softwareprodukte ausprobiert werden, ohne dass die Systemeinstellungen des realen Betriebssystems verändert werden. Auch Updates und Hardwaretreiber (in eingeschränktem Umfang) werden so vor dem Echt-Betrieb auf ihre Funktionsfähigkeit getestet.

Die Virtualisierung mittels Virtual Machine Monitors (VMM) arbeitet mit einem Hypervisor, der zwischen den einzelnen virtuellen Maschinen mit ihren Gastsystemen und dem Host-Betriebssystem umschaltet. Der Hypervisor ist eine Art Überwachungs- und Monitoring-Instanz. Die realen Hardwareressourcen werden dabei möglichst intelligent verteilt, virtualisiert und ggf. auch emuliert und somit werden Gastsystemen komplette Rechner mit CPU, RAM, Laufwerken und I/O-Schnittstellen dargestellt. Emulation bedeutet das Nachahmen der Eigenschaften von Original-Hardware, Simulation das Nachbilden der Original-Hardware und deren Verhalten, Virtualisierung das Nutzen der Original-Hardware über eine Zwischenschicht, wobei Teile auf der Original-Hardware ausgeführt werden. Nach diesem Prinzip arbeiten VMware Workstation[31] und Server, Microsoft Virtual PC[32] und Server, sowie Parallels Workstation[33] und Innotek Virtual Box[34].

Bei Virtualisierung auf Betriebssystemebene wird kein kompletter Rechner dargestellt, sondern nur eine Laufzeitumgebung. Dabei nutzen die Gastsysteme den Kernel des Hostsystems und sind bei der Auswahl der Gastsysteme eng an das Hostsystem gebunden, da keine anderen Treiber oder Kernel genutzt werden können, als die, die der Host anbietet. Die systemnahe Integration bietet so die Möglichkeit, sehr viele gleichartige Gastsysteme auf einem Hostsystem laufen zu lassen und wird gerne von Webprovidern genutzt. Ein Beispiel dafür ist "jails" von BSD [BSD07].

Bei der sogenannten Paravirtualisierung kann dagegen ein weiteres Betriebssystem inklusive Kernel gestartet werden, wobei nicht die komplette Hardware virtualisiert wird, sondern das Gastsystem über eine Verwaltungsebene auf die Ressourcen des Hostsystems zugreift. Inzwischen ist das Open Source-Produkt Xen[35] von der Firma Citrix erweitert worden, so dass es möglich ist, das Microsoft Windows-Betriebssystem Server 2003 auf dem

[31] http://www.vmware.com/de/products/ws/ (kostenpflichtig)
http://www.vmware.com/de/products/server/

[32] http://www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx
http://www.microsoft.com/germany/virtualserver/default.mspx

[33] http://www.parallels.com/products/workstation/

[34] http://www.virtualbox.org/

[35] http://www.cl.cam.ac.uk/research/srg/netos/xen/

Citrix XenServer v4[36] zu betreiben. Die kostenfreie eingeschränkte Version Express Edition kann im Internet heruntergeladen werden.

Weitere Spielarten der Virtualisierung sind zum einen die komplette Hardware-Simulation, bei der auch auf dem Gast eine andere Hardwareplattform (z.B. PowerPC) laufen kann, als auf dem Hostsystem mit beispielsweise x86-Architektur. Zum anderen die Anwendungsvirtualisierung, wobei einzelne Anwendungen in einer virtuellen Umgebung ausgeführt werden und vom lokalen Betriebssystem getrennt sind. Anwendungsvirtualisierung findet statt bei Microsoft Softgrid[37] und Sun Java Virtual Machine[38].

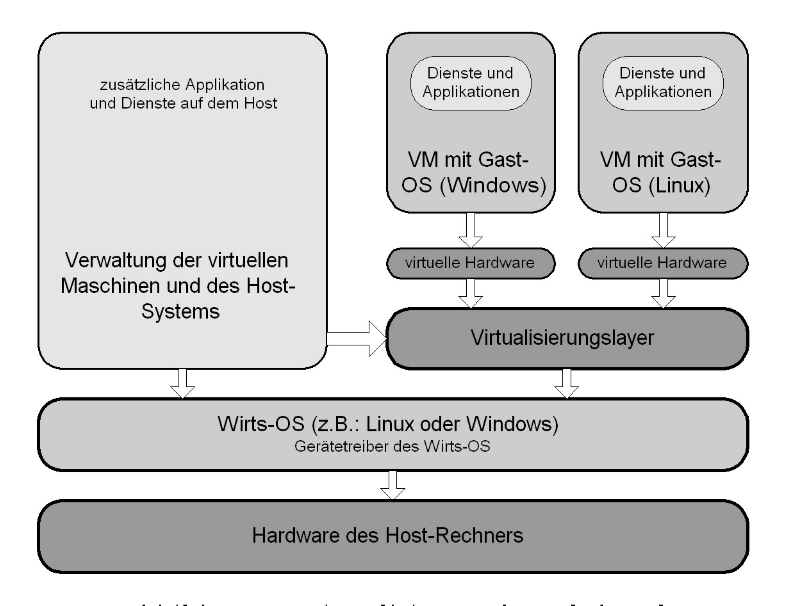


Abbildung 8.1 Virtualisierungslayer [Ahn07]

Folgende Punkte sprechen für den Einsatz einer Virtualisierung mittels einer VMM-Lösung:

- Es müssen komplette Rechner virtualisiert werden, die sowohl Windows, als auch Linux als Betriebssystem besitzen.
- Es kommt ausschließlich x86-Hardware bei Gast und Host zum Einsatz.
- Das Gesamtsystem muss performant sein.

Aus den oben genannten Produkten wurde sich für den Einsatz von VMware Server in der Version 1.0.4 entschieden. Ausschlaggebend für die Entscheidung waren:

- Kann auf mehr Linux- und Windows-Host- und Gastbetriebssystemen ausgeführt werden als alle anderen derzeit verfügbaren Produkte zur Server-Virtualisierung.
- Unterstützt Virtual SMP mit zwei Prozessoren, so dass eine einzige virtuelle Maschine mehrere physische Prozessoren nutzen kann.
- Die einfache Möglichkeit vorhandene virtuelle Maschinen auf eine Enterprise Lösung (VMware ESX) zu migrieren.
- Kann mit VMware VirtualCenter gemanagt werden, um die Infrastruktur von einer zentralen Management-Konsole aus effizient bereitzustellen, zu überwachen und zu managen.

---

[36] http://www.citrixxenserver.com/products/Pages/XenExpress.aspx

[37] http://www.microsoft.com/systemcenter/softgrid/default.mspx

[38] http://www.java.com/de/download/manual.jsp

- Kann das virtuelle Maschinenformat von VMware und Microsoft sowie Symantec LiveState Recovery-Images mit VMware Converter (früher VMware Importer) öffnen.
- Vorhandene jahrelange Erfahrung mit VMware Produkten.

8.1 Allgemeines

Das nachfolgende Schaubild zeigt an einem Beispiel, wie sich das Speichern eines Dokuments innerhalb einer virtuellen Maschine darstellt und an welcher Stelle der reale Rechner ins Spiel kommt.

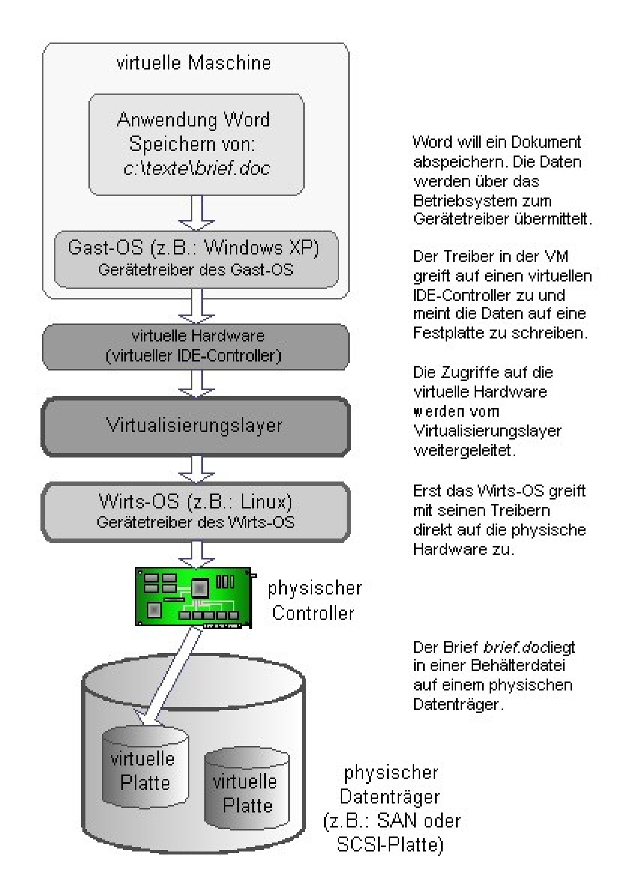


Abbildung 8.2 Zugriff einer VM auf Hardware [Ahn07]

8.1.1 Testanlage

Neben den vorangestellten Ausführungen zum Einsatz von virtuellen Maschinen – als Terminalserver oder Endgeräte am Arbeitsplatz des Administrators – bietet sich die Verwendung im Rahmen einer Testanlage an. Das Haupteinsatzgebiet der Testanlage in der KVMYK liegt im Test von neuen Verfahren. Des Weiteren können dort größere und umfangreichere Änderungen am gesamten Netzwerk und dem Active Directory getestet werden. Dabei liegen die Anforderungen nur beim ausführlichen Test der Auswirkungen der Änderungen von Einstellungen sowie der generellen Lauffähigkeit und Eignung von neuen Softwareverfahren. Keine Rolle spielen hier Aspekte wie die physikalische Änderung des Netzwerks, z.B. durch Austausch der aktiven Komponenten (Switches oder Hubs) sowie Messungen zur Performanz der Testanlage.

Es müssen also alle Server aus der produktiven Umgebung virtualisiert werden, die zum Test von Softwareverfahren gebraucht werden. Folgende Server sind dazu notwendig:

- Domänencontroller
- Fileserver
- Terminalserver

Neben diesen grundsätzlich immer benötigten Server – im Folgenden als Basisserver bezeichnet – gibt es auch eine Menge weiterer Server, die je nach Testszenario hinzugefügt werden müssen, wie z.B.:

- Mailserver (Test von E-Mail)
- Printserver (Test von Druckfunktionalität)
- Datenbankserver
- Imageserver
- Proxyserver oder Firewall (z.B. bei Zugriffen ins Internet)
- AS/400
- Faxserver

Nicht immer ist es mit vertretbarem Aufwand oder technisch überhaupt möglich, einen der oben genannten Server zu virtualisieren. Soll beispielsweise ein neues Verfahren auf der AS/400 getestet werden, scheidet eine Virtualisierung aus und es steht nur der Test auf dem produktiven System zur Wahl. Soll aber eine neue Terminalemulation für den Zugriff auf die AS/400 getestet werden, reicht es in solchen Fällen oft aus, einen Zugriff aus der Testanlage heraus auf die realen Systeme zu ermöglichen. Dieser Zugriff kann durch eine Firewall geregelt werden, die das Testsystem von der produktiven Umgebung abkoppelt und nur den selektiven Zugriff auf dedizierte Hosts zulässt.
Natürlich werden in einer solchen Umgebung auch Clients benötigt. Dies kann sowohl mit virtuellen Maschinen als auch realen Computern bewerkstelligt werden.

In der nachfolgenden Skizze ist der Zugriff aus dem Netzwerk "Kreishaus" für Administratoren-Arbeitsplätze über eine Firewall auf das Management-Netzwerk VMnet0 der Host-Maschinen *VMTHOST01* und *VMTHOST02* möglich. Ein direkter Zugriff aus dem Kreishaus in das Netzwerk VMnet2 ist nicht gestattet. Aus dem Test-Netzwerk VMnet2 ist dagegen ein selektiver Zugriff über eine Firewall auf das Netzwerk "Kreishaus" möglich. Dort ist als mögliches Ziel die AS/400 abgebildet.

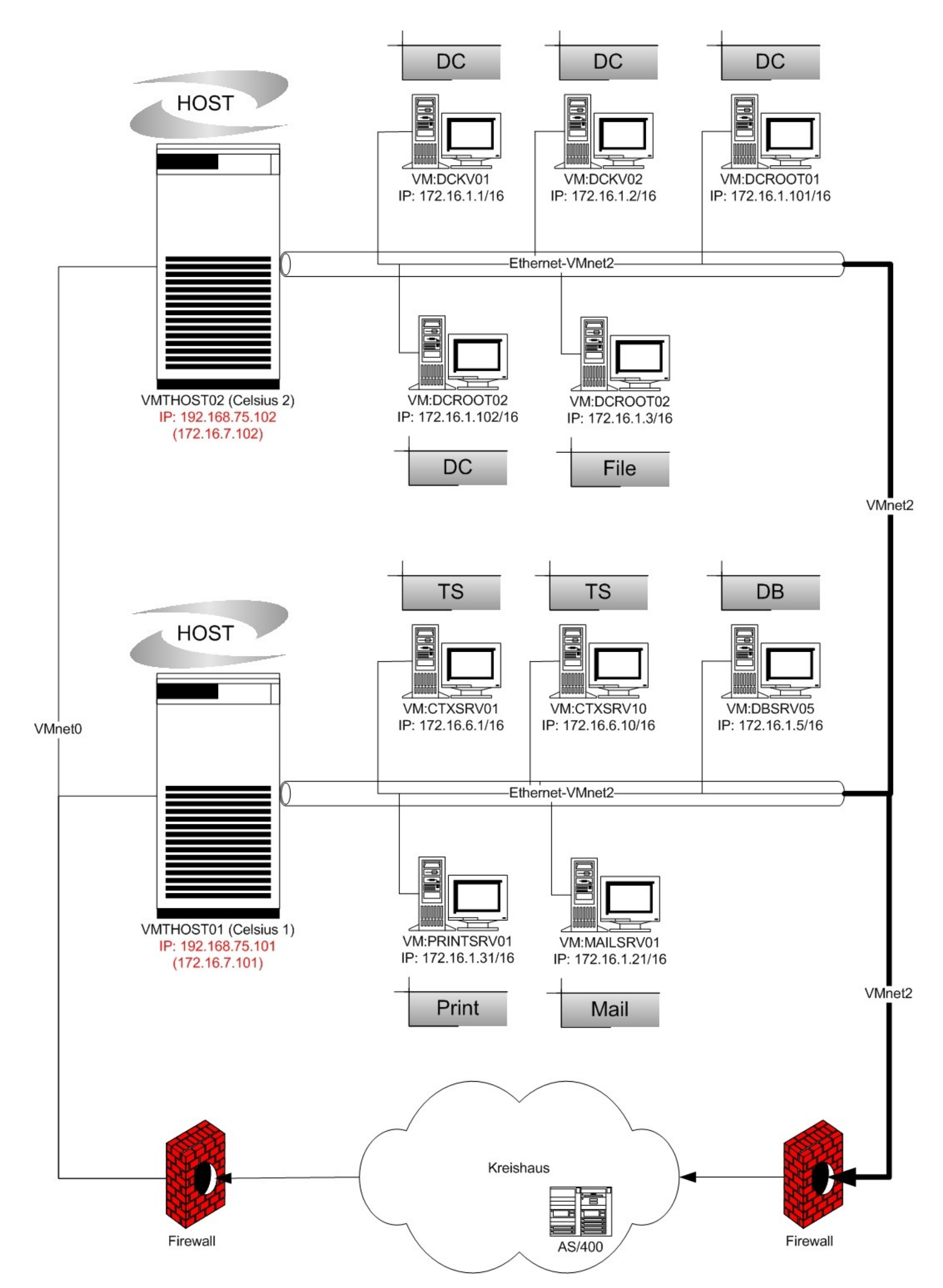


Abbildung 8.3 Testanlage Schemazeichnung

8.1.2 Produktivumgebung

Für den Einsatz in der produktiven Umgebung wird ebenfalls der kostenfreie VMware Server eingesetzt. Diese Software wird auf einem dedizierten Host namens *VMHOST01* installiert, der in der Folge nur noch virtuelle Maschinen hosted und keinerlei sonstige Funktionen wahrnimmt. Dieser stellt dann bei Bedarf virtuelle Terminalserver oder sonstige virtuelle Server zur Verfügung. Der *VMHOST01* ist ein HP Server, der die unten dargestellten Eckdaten aufweist und bzgl. seiner verwendeten Komponenten (Netzteil, Lüfter, Festplatte, Netzwerkkarte) auf Redundanz und Ausfallsicherheit optimiert ist.

- HP Proliant DL 380 G 3
- 2x XEON 3,2 GHz
- 4 GB RAM

- 2x136GB im RAID1-Verbund
- 4x18GB im RAID5-Verbund
- DVD-ROM
- 2x 10/100/1000TX Netzwerkkarte
- 1x 1000FX Netzwerkkarte

Der Server ist direkt im Backbone des aktiven Netzes mittels Glasfasertechnik über 1GBit/s angebunden. Dort laufen derzeit schon drei produktive virtuelle Server und bei Bedarf können noch bis zu zwei weitere Server auf dem *VMHOST01* gehosted werden.
Am Arbeitsplatz des Administrators ist auch VMware Server vorgesehen. Allerdings ist an manchen dieser Arbeitsstationen bereits die Version VMware Workstation im Einsatz. Dies wird in der Folge gleichfalls Berücksichtigung finden. Die Arbeitsstationen der Administratoren sind auch leistungsfähiger als ein Standardarbeitsplatz und können daher ohne zusätzliche Kosten virtuelle Maschinen beherbergen.

Zusammenfassung

Es sind zwei große Einsatzgebiete der Virtualisierung zu unterscheiden: Einerseits der Einsatz in der produktiven Umgebung mit den Anforderungen der schnellen und universellen Verfügbarkeit sowie dem möglichen Einsatz von virtuellen Endgeräten am Arbeitsplatz eines Administrators. Auch hier spielt Performanz der virtuellen Maschinen eine Rolle.
Andererseits die Virtualisierung einer kompletten Testanlage. Hier spielt Performanz eine untergeordnete Rolle. Wichtig ist dort in erster Linie die komplette technische Nachbildung der produktiven Echtumgebung, um ausführliche und realitätsgetreue Testergebnisse zu reproduzieren.

8.2 Planung der Testanlage

Die Testanlage soll auf zwei schon vorhandenen Computern installiert werden. Diese Computer waren ehemals Teil der Testanlage zur Vorbereitung der Migration. Dort wurde der Test der Kompatibilität der Softwareverfahren bezüglich ihrer Terminalservertauglichkeit ausgeführt sowie eine Testmigration von Windows NT auf Windows Active Directory durchgeführt. Dabei handelt es sich um zwei 4 Jahre alte Power-Workstations Celsius 670 von FSC. Diese bestanden aus je folgenden Komponenten:

- 2x Intel Pentium 4 XEON 2,0 GHz
- 2 GB RAM
- 120 GB IDE Festplatte
- DVD-ROM
- onboard 10/100 TX Netzwerkkarte

Da diese Computer in der Migration-Testanlage jedoch leichte Schwächen in der Performanz und eine große Auslastung der Kapazitäten von RAM und Festplatte zeigten, wurden die Rechner mit teils vorhandener, teils neu gekaufter Hardware wie folgt aufgerüstet:

- Celsius 1 mit 2x Intel Pentium 4 XEON 2,0 GHz
- 2,5 GB RAM
- 120 GB IDE Festplatte (System)
- 500 GB SATA Festplatte (Backup)
- 2x250 GB SATA im RAID 0 Verbund (VMdaten)
- DVD-ROM

- onboard 10/100 TX Netzwerkkarte (VMnet0)
- PCI 10/100/1000 TX GigaBit Ethernet Netzwerkkarte (VMnet2)

Der Rechner Celsius 2 ist identisch mit Celsius 1, hat allerdings 3 GB RAM.
Einerseits um den leichten Mangel an RAM auszugleichen, andererseits um die prinzipielle Machbarkeit zu zeigen, wird auf dem Rechner Celsius 2 das Betriebssystem Windows Server 2003 Standard und auf dem Celsius 1 das Betriebssystem OpenSuSE 10.3 verwendet. Beide Betriebssysteme werden in der 32-Bit-Version installiert, da VMware Server nicht auf einem 64-Bit System läuft. Auf OpenSuSE ist die Installation der 64-Bit Version möglich, dort muss dann aus Gründen der Kompatibilität das Paket "compat" installiert werden. Da es die Software VMware Converter ausschließlich als Windows-Version gibt, werden nicht beide Server mit dem kostenfreien[39] OpenSuSE 10.3 installiert. Die Software VMware Converter wird folglich auf dem Rechner Celsius 2 installiert.

In der fertig zu stellenden Anlage müssen immer die Basisserver aus der Produktivumgebung abgebildet werden.

- 2 Domänencontroller der übergeordneten Domäne (*Root*)
- 2 Domänencontroller der aktuellen Domäne (*KV*)
- Fileserver
- 2 Terminalserver (Desktop und Sonderanwendungen)
- Datenbankserver (soweit von den Terminalservern unter Citrix benötigt)

Je nach Testszenario kommen weitere Server hinzu, die nach Möglichkeit ebenfalls mit der verfügbaren Hardware abgebildet werden sollen.
Da beim Imagen sehr viel I/O-Performanz (Netzwerk und Festplatte) von Nöten ist, werden die beiden Imageserver auf einer zusätzlichen dritten Maschine abgebildet. Dies ist ein normaler Desktop-Rechner FSC Esprimo E 5600 mit AMD Sempron 3000, 1 GB RAM und 40GB Festplatte und dem Hostnamen *VMMANAGER*. Auf diesem Rechner läuft ebenfalls eine Instanz VMware Server mit dem virtuellen Server *SCOUTNG*. Die in der Echtumgebung ebenfalls vorhandenen Dienste wie Proxy, Intranet und Firewall werden ebenfalls auf dieser Maschine abgebildet, da sie von nachrangiger Bedeutung sind und nicht in allen Fällen in der Testumgebung benötigt werden.

[39] Wenn die Version aus dem Internet heruntergeladen wird http://software.opensuse.org

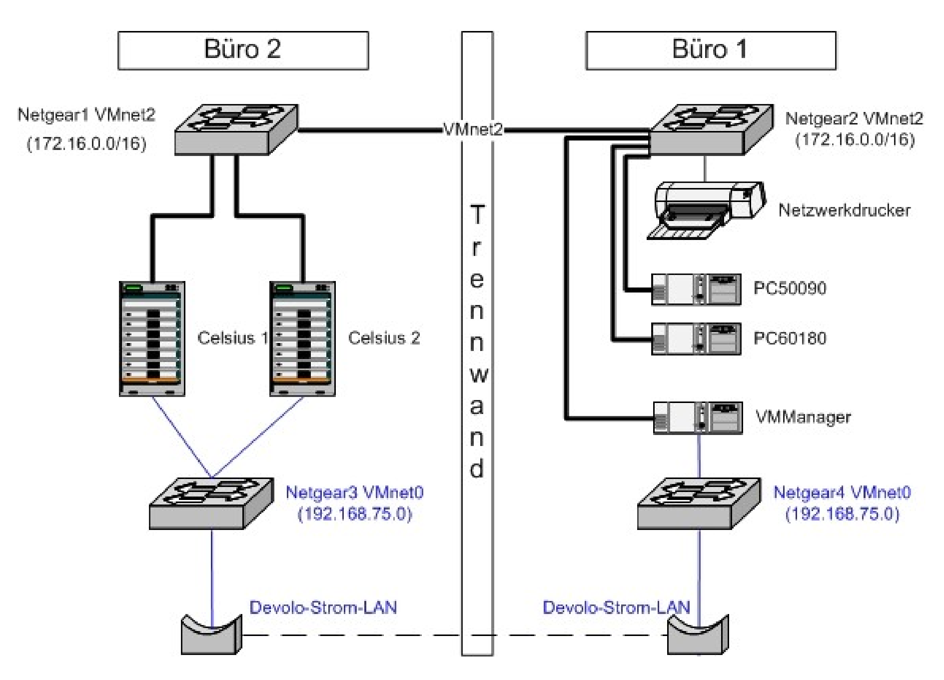


Abbildung 8.4 Aufbau der Testanlage im Büro der Fa. Neton

Die obige Zeichnung zeigt den Aufbau der Testanlage im Büro der Fa. Neton. Die Anlage ist dabei in zwei Teilen auf die beiden Büros verteilt, die durch eine Wand voneinander getrennt sind. Das Netz VMnet2 ist durch Kabel, die über Patchfelder führen, direkt verbunden. Der Management-Netzwerk VMnet0 ist durch zwei Devolo dLan-Adapter verbunden, die eine Verbindung über das 230 V Stromnetz herstellen. Die Geschwindigkeit dieser Adapter beträgt maximal 14 MBit/s. An den Rechner *VM Manager* ist noch ein externer Datenträger zur Speicherung und Datensicherung angeschlossen.

8.3 Installation der Testanlage

Beide Server werden in ähnlicher Weise vorbereitet. Die Konfiguration der beiden Maschinen ist unten in der Tabelle dargestellt.

	Celsius 1	Celsius 2
CPU	2x XEON 2,0 GHz	2x XEON 2,0 GHz
RAM	2,5 GB	3,0 GB
Hostname	VMTHOST01	VMTHOST02
IP-Adresse (VMnet0) [10/100 TX]	192.168.75.101	192.168.75.102
IP-Adresse (VMnet2) [10/100/1000 TX]	172.16.7.101	172.16.7.102
Festplatte 1 (500 GB) [1x 500 GB]	/dev/sda1 (10 GB; FS: ext3) Mountpoint: /	(8 GB; FS: ntfs) C: (BOOT)
	/dev/sda2 (Rest; FS: ext3) Mountpoint: /install	(Rest; FS: ntfs) D: (INSTALL)

Festplatte 2 (500 GB) [2x 250 GB RAID0]	/dev/sdb1 (500 GB; FS: ext3) Mountpoint: /vmdaten	(500 GB; FS: ntfs) F: (VM Daten)
Festplatte 3 (500 GB) [1x 500 GB]	/dev/sdc1 (10 GB; FS: ext3) Mountpoint: /backup	(Rest; FS: ntfs) F: (Backup)
DVD-ROM	/dev/sr0	E:
Betriebssystem	OpenSuSE 10.3	Windows Server 2003

Tabelle 8.1 Konfiguration der Testanlagen-Server

Dabei ist das Management-Netzwerk über VMnet0 mit maximal 100 MBit/s und das produktive Netzwerk über VMnet2 mit maximal 1000 MBit/s erreichbar. Im VMnet0 werden zwei Netgear 10/100 TX Miniswitches mit 8 Ports eingesetzt. Im VMnet2 kommen zwei Netgear 8 Port Gigabit Smart Switches GS108T zum Einsatz. Die Devolo-Strom-LAN-Adapter werden nur im Büro der Fa. Neton benutzt und sind im späteren Betrieb der Testanlage nicht mehr nötig.
Auf der Festplatte 1 erfolgt die Installation des Betriebssystems in eine kleine Partition von 8-10 GB Größe, der Rest steht zur Ablage von Installationsdateien o. ä. zur Verfügung. Bei dem Linux-Rechner kommt das Dateisystem (FS) "ext3" zum Einsatz, bei dem Windows-Rechner das Dateisystem "NTFS".
Auf der Festplatte 2, die wegen der besseren Performanz aus einem RAID0-Verbund von zwei Festplatten besteht, werden die aktiven virtuellen Maschinen gespeichert. Dort besteht folgende Verzeichnisstruktur aus zwei Verzeichnissen, von denen das Verzeichnis *vmachine* unter dem Namen "vmachine" freigegeben ist:

- *vmachine* (produktive virtuelle Maschinen)
- *vmachinetest* (virtuelle Testmaschinen)

Auf der Festplatte 3 werden Backups der virtuellen produktiven Maschinen gespeichert, die jeden Abend automatisch dorthin gespeichert werden. Dazu wird das Verzeichnis *vmachinebackup* genutzt, welches über die Freigabe "vmbackup" über das Netzwerk erreichbar ist. Alle Freigaben sind durch entsprechende Rechte vor dem unbefugten Zugriff geschützt. Lediglich der Administrator hat Zugriff darauf. Das Script *vm-do-backup.cmd* sichert jeweils das Verzeichnis *vmachine* der beiden Hosts über kreuz in die jeweilige Freigabe "vmbackup". Das Script läuft um 20:00 Uhr auf dem *VMTHOST02* und kopiert den Inhalt der produktiven Maschinen von *VMTHOST01* auf *VMTHOST02* und umgekehrt. Dabei wird pro Wochentag ein gleichlautendes Verzeichnis angelegt, in das die Sicherungen kopiert und bei Bedarf überschrieben werden.
Die Scripte dazu sind im Anhang abgedruckt.

8.3.1 Hostmaschinen

Die Installation der beiden Hostmaschinen Celsius 1 und Celsius 2 ist relativ einfach und schnell vollzogen. Beide Betriebssysteme, also Windows Server 2003 Standard als auch Novell OpenSuSE 10.3 werden mit den Default-Einstellungen installiert.
Für OpenSuSE heißt das ein minimales Grafisches Grundsystem mit Netzwerkunterstützung. Zusätzlich wird noch das Paket "Samba" ausgewählt. Dieses Paket bietet die Unterstützung für Windows und implementiert das SMB-Protokoll unter Linux. Das SMB-Protokoll ist das Kommunikationsprotokoll, das Windows-Rechner in Netzwerken benut-

zen, um den Zugriff auf Datei- und Druckdienste abzuwickeln. Nach entsprechender Konfiguration erscheint dann der *VMTHOST02* nachher wie ein Windows-Rechner im Netzwerk und so kann das Script *vm-do-backup.cmd* aus dem letzten Kapitel den Zugriff von dem Rechner *VMTHOST02* auf den *VMTHOST01* einfach gewährleisten. Die Netzwerkunterstützung wird wie in der oben aufgelisteten Tabelle eingerichtet. Es ist empfehlenswert, die Firewall im ersten Schritt zu deaktivieren bzw. beide Interfaces der Schnittstelle "intern" zuzuweisen. Wenn die benötigten Kommunikationsports der verschiedenen Softwarekomponenten bekannt sind, kann die Firewall auch nachträglich aktiviert werden. Es existieren allerdings keine fertigen Firewall-Regeln für das VMware Produkt, so dass diese manuell erstellt werden müssen.
Ein weiterer wichtiger Aspekt ist, dass sowohl ein Kompiler (z.B. gcc) und die Linux-Kernelquellen mitinstalliert werden. Diese werden später bei der Installation des VMware Servers benötigt, da Teile des VMware Servers erst zur Laufzeit und in Abhängigkeit vom der Version des Kernels kompiliert werden.
Die grafische Unterstützung wird eigentlich nicht benötigt, aber unter einer grafischen Konsole sind einige Aufgaben bequemer zu bearbeiten. Um die Ressourcen zu schonen, wird beim Systemstart nur der Textmodus gestartet. Bei Bedarf kann dann in den grafischen Modus umgeschaltet werden. Für die vollständige Installation des VMware Servers wird noch ein Webserver benötigt und daher wird das Paket "Apache" ebenfalls installiert. Anschließend kann der VMware Server installiert werden. Unter Windows ist die Installation relativ schnell abgeschlossen. Durch einen Doppelklick auf die Datei *VMware-server-installer-1.0.4-56528.exe* werden alle benötigten Komponenten abgefragt und bei Bedarf installiert. Im Anschluss wird auch das Webinterface automatisch konfiguriert und ein Zertifikat für den verschlüsselten Zugriff über das Protokoll https erzeugt.
Unter Linux ist die Installation geringfügig aufwändiger. Dort wird das Paket *VMware-server-1.0.4-56528.i386.rpm* installiert und die Steuerung wird durch Abfragen geregelt. Anschließend muss für die Installation des Webinterfaces noch der gepackte Tarball[40] *VMware-mui-1.0.4-56528.tar.gz* entpackt und installiert werden. Anschließend erfolgt die Konfiguration der Software durch Aufruf des Shell-Scripts */usr/bin/vmware/vmware-config.pl*. Im Anschluss daran wird auf das Wunsch automatisch das Script *vmware-config-mui.pl* zur Konfiguration des Webinterfaces aufgerufen.
Während der Installation muss eine Seriennummer eingegeben werden. Diese ist kostenlos auf der Homepage von VMware erhält. Für die Windows- oder Linux-Variante müssen unterschiedliche Seriennummern eingegeben werden, die nicht zueinander kompatibel sind. Deshalb müssen für Windows und Linux jeweils eigene Seriennummern registriert werden.
Die Netzwerkverbindungen werden entsprechend VMnet0 bzw. VMnet2 benannt. Die Nachfrage für die Netzwerkunterstützung während der Linux-Installation des VMware Servers wird bejaht. Diese Frage erscheint unter Windows nicht, dort wird sie automatisch eingerichtet. Automatisch heißt hier, dass zwei zusätzliche Netzwerkschnittstellen namens "VMware Network Adapter VMnet1" und "VMware Network Adapter VMnet8" eingerichtet werden und das VMware Bridge Protocol an alle vorhandenen Adapter gebunden wird. Das VMware Bridge Protocol bietet die Möglichkeit, für die virtuellen Maschinen über die vorhandenen physikalischen Netzwerkadapter zu kommunizieren. Über das Programm Manage Virtual Networks können alle Schnittstellen konfiguriert werden.

---

[40] *tar* ist der Name eines im Unix-Umfeld sehr geläufigen Archivierungsprogramms. Die Endung *gz* weist darauf hin, der Tarball gezipped ist.

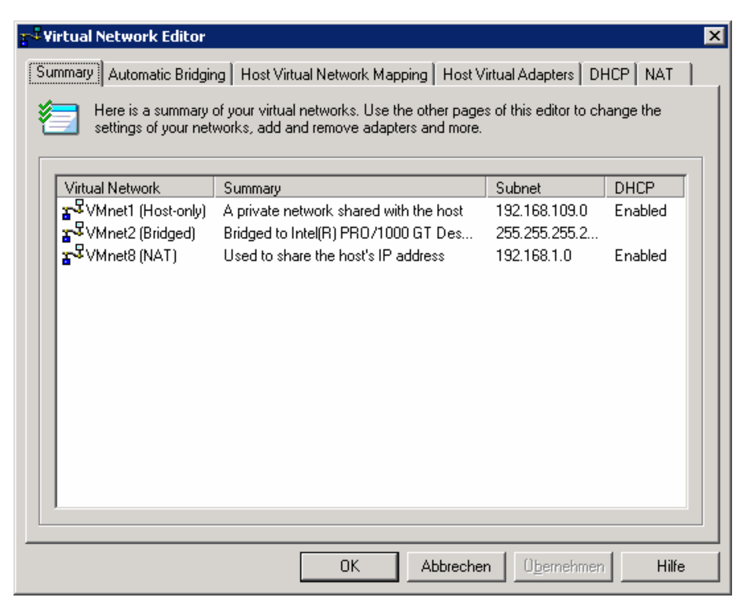


Abbildung 8.5 Manage Virtual Networks

Der Adapter VMnet0 ist immer der Default-Adapter, der für das Bridging über einen realen Netzwerkadapter benutzt wird. Bridging bedeutet hier, dass die virtuelle Maschine über diesen Adapter wie ein zusätzlicher Computer auf demselben physischen Netzwerk wie der Host-Rechner erscheint. Damit kann die virtuelle Maschine sämtliche auf diesem Netzwerk verfügbaren Dienste in Anspruch nehmen. VMnet1 ist der sogenannte Host-only-Adapter und bietet virtuellen Maschinen nur die Möglichkeit mit dem Host-Rechner in Verbindung zu treten. Der Adapter VMnet8 kann über NAT einen vorhandenen Adapter des Host-Rechners nutzen und teilt sich mit ihm die IP-Adresse des Host-Rechners. Die Default-Einstellungen werden geändert und es wird nur VMnet0 bzw. VMnet2 genutzt, VMnet1 und VMnet8 können deaktiviert werden, um Ressourcen zu schonen. VMnet2 wird dabei an den leistungsfähigeren Adapter gebunden und für das VMware Bridge Protocol zugelassen. Sämtliche anderen Bindungen könnten entfernt werden, da sie im Grunde nicht benötigt werden. Dann würde allerdings die tägliche Sicherung der Gastmaschinen über das Script *vm-do-backup.cmd* ausschließlich über das langsamere Netzwerk VMnet0 abgewickelt. Erfolgt in einer produktiven Umgebung die Sicherung der Maschinen beispielsweise auf einen zentralen Backupserver im Netz, ist es sinnvoll an VMnet2 die "Datei- und Druckerfreigabe für Windows-Netze" zu binden und dem Host ebenfalls eine gültige IP-Adresse aus diesem Netz zu vergeben. Gleiches gilt natürlich auch für einen Linux-Rechner, wo die Netzwerkschnittstelle und der SMB-Daemon entsprechend konfiguriert sein müssen.

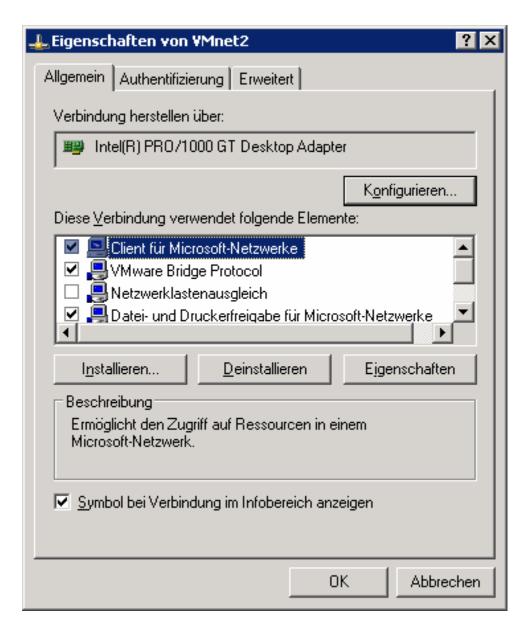


Abbildung 8.6 Eigenschaften von VMnet2

Der Adapter VMnet0 wird ausschließlich vom Host-Rechner genutzt und bekommt daher keinerlei Bindung an das VMware Bridge Protocol und wird nur für das Management der Host-Maschinen genutzt. Nur so kommt die in Abbildung 8.3 und Abbildung 8.4 gezeigte Trennung der beiden Netze zustande.
In diesem Punkt unterscheiden sich auch die beiden Server-Konsolen für die Steuerung und Verwaltung des Servers. Unter Windows existiert der Menüpunkt "Host/Virtual Network Settings" und die Netzwerkunterstützung kann dort grafisch vorgenommen werden. Diesen Komfort bietet die Konsole unter Linux nicht. Dort muss die Konfiguration in einer Shell über das Script *vmware-config.pl* vorgenommen werden. Sind also dort Änderungen zu machen, ist ein Zugriff auf die Shell des Linux-Rechners erforderlich. Dies kann z.B. mit *putty* in einer SSH-Sitzung komfortabel erledigt werden. Es ist deshalb auch darauf zu achten, dass der SSH-Daemon gestartet, konfiguriert und in der Firewall entsprechend freigeschaltet ist.

Einen weiteren Unterschied zwischen den Betriebssystemen gibt es in der Konfiguration der VMware-Verwaltungskonsole. Standardmäßig wird diese Remote-Konsole auf TCP-Port 902 gebunden. Im Falle von OpenSuSE ist dieser Port bereits für "ideafarm-chat" reserviert, so dass während der Installation der TCP-Port 904 vorgeschlagen wird. Dies ist bei der späteren Verbindung über die Remote-Konsole zu beachten, da der Port zusätzlich anzugeben ist, wenn er vom Default-Wert abweicht. Soll trotzdem der Port 902 benutzt werden, muss die Datei */etc/services* angepasst werden. Der an der Stelle 902 vorhandene Eintrag wird entfernt und durch "vmware-authd 902/tcp #VMware-authd" ersetzt.



Abbildung 8.7 Anmeldung an der VMware Console

Sollen die späteren Gastmaschinen nach einem Neustart des Hosts automatisch gestartet werden, muss in der Konfiguration der Gastmaschinen ein Account zugewiesen werden, mit dessen Berechtigungen der Gast ausgeführt wird. Bei Windows-Hosts wird dort "Local System" eingetragen. Diesen Menüpunkt gibt es in der Linux-Konsole der VMware nicht. Dort laufen die Gastmaschinen mit den Berechtigungen des Besitzers der zugehörigen Konfigurationsdatei mit der Dateiendung *vmx*. Dies ist im Default der Benutzer "root" der lokalen Maschine. Für die spätere Sicherung der Gastmaschinen muss die Berechtigung entsprechend angepasst werden, damit der Benutzer Administrator die Gastmaschinen über die Freigabe sichern und ggf. zurücksichern darf.

8.3.2 Gastmaschinen

Die eigentliche Installation von virtuellen Maschinen unterscheidet sich nur wenig von realen Systemen. Der erste Schritt ist das Zusammenstellen einer virtuellen Maschine. Das geschieht nach dem Baukastenprinzip assistentengesteuert über die VMware Server Console. Nach dem Aufruf startet man über "File/New/Virtual Machine" oder direkt über die Tastenkombination "Strg-N" den Assistenten. Nach der Entscheidung, ob eine typische oder eine benutzerdefinierte Maschine erstellt werden soll, kann das Zusammenstellen beginnen. Es ist in den meisten Fällen sinnvoll die benutzerdefinierte Variante (Custom) zu wählen, da ansonsten nur wenig Einfluss auf die Konfiguration der Maschine besteht.

Quellsystem (physikalisch)		Zielsystem (virtuell)	
Prozessor:	1	Prozessor:	1
RAM:	1024 MB	RAM:	384 MB
Festplatte 1 (SCSI):	36 GB	Festplatte 1 (SCSI):	36 GB
Festplatte 2 (SCSI)	nicht vorhanden	Festplatte 2 (SCSI)	2 GB
Netzwerkkarte 1:	Ethernet 10/100/1000 TX	Netzwerkkarte 1:	Ethernet 10/100/1000 TX
CD- / DVD-ROM:	vorhanden	CD- / DVD-ROM:	vorhanden

Tabelle 8.2 Reale und virtuelle Hardwaredefinitionen

Die obige Tabelle zeigt die Konfiguration eines realen bzw. virtuellen DCs mit dem Namen *DCROOT01* und dem Betriebssystem Windows Server 2003 Standard. Dabei fällt auf, dass die virtuelle Maschine weniger RAM hat als die reale. Das liegt einerseits daran, dass moderne Serverhardware in der kleinsten Konfiguration schon 1 GB RAM besitzt. Andererseits kommt es in der virtuellen Maschine nicht auf Performanz an, so dass das zur Verfügung gestellte RAM verkleinert wird und die Funktionalität trotzdem gewährleistet bleibt. Im Gegenzug wird eine zweite kleine Festplatte hinzugefügt, auf der die Auslagerungsdatei platziert wird. So erhält die Auslagerungsdatei eine eigene Festplatte, was sich positiv auf die Leistung auswirkt. Das Netzwerk der virtuellen Maschine wird auf VMnet2 konfiguriert.

Nachdem der Assistent beendet ist, speichert dieser die Konfiguration in einer Datei mit der Endung *vmx* ab. Diese Datei liegt im Ascii-Format vor und hat für das o. g. Beispiel folgenden Inhalt:

```
config.version = "8"
virtualHW.version = "4"
memsize = "384"
MemAllowAutoScaleDown = "FALSE"
MemTrimRate = "30"
displayName = "DCROOT01"
guestOS = "winnetstandard"
numvcpus = "1"
tools.remindInstall = "FALSE"
ethernet0.present = "TRUE"
ethernet0.addressType = "generated"
ethernet0.connectionType = "custom"
ide1:0.present = "TRUE"
ide1:0.autodetect = "TRUE"
ide1:0.filename = "auto detect"
ide1:0.deviceType = "cdrom-raw"
scsi0.present = "TRUE"
scsi0.virtualDev = "lsilogic"
lsilogic.noDriver = "FALSE"
priority.grabbed = "normal"
priority.ungrabbed = "normal"
ethernet0.generatedAddressOffset = "0"
scsi0:0.redo = ""
ethernet0.generatedAddress = "00:0c:29:00:51:01"
uuid.location = "56 4d 28 b7 7b 7e 9e 44-b9 e9 e2 5a 38 00 51 01"
uuid.bios = "56 4d b9 06 95 f3 7a 62-cf 0f 43 31 6b 00 51 01"
Ethernet0.vnet = "VMnet2"
ide1:0.startConnected = "FALSE"
```

```
tools.syncTime = "FALSE"
workingDir = "."
```

Ab diesem Zeitpunkt kann die virtuelle Hardware prinzipiell gestartet werden und unterscheidet sich dann kaum noch von einer realen Maschine. Es kann also eine Installation von einer CD-ROM erfolgen oder es kann die gleiche Imagetechnik wie bei realen Computern angewendet werden.

Die folgenden Konventionen werden für die Erstellung und Benennung von virtuellen Maschinen festgelegt:

(1) Der Name (Virtual Machine Name) der virtuellen Maschine in der VMware Console entspricht dem tatsächlichen Hostnamen des späteren Rechners, z.B. *DCROOT01*.
(2) Die Maschine wird in einem Verzeichnis (Location) erstellt, das so heißt wie die Maschine, z.B. *DCROOT01*.
(3) Die Konfigurationsdatei der virtuellen Maschine mit der Endung *vmx* heißt so wie der Hostname der Maschine, z.B. *DCROOT01.vmx*.
(4) Die Datei zur Aufnahme von Metainformationen mit der Endung *vmsd* heißt so wie der Hostname der Maschine, z.B. *DCROOT01.vmsd*.
(5) Die virtuellen Festplatten werden nach folgendem Schema benannt: *Hostname_laufendeNr.vmdk*, z.B. *DCROOT01_1.vmdk*.
(6) Der Haken bei "Allocate all disk space now" wird entfernt.
(7) Der Haken bei "Split disk into 2 GB files" wird gesetzt.
(8) Für Server werden nur virtuelle SCSI-Festplatten verwendet, für Endgeräte nur IDE-Laufwerke.
(9) Diskettenlaufwerke werden nur nach Bedarf hinzugefügt und der Haken "Connect at power on" wird deaktiviert.
(10) Bei einem CD-ROM wird der Haken "Connect at power on" deaktiviert.
(11) Bei den Netzwerkadaptern wird immer das konkrete Interface (VMnet0 oder VMnet2) eingestellt und nicht "bridged".
(12) Die beiden letzten Oktette der späteren IP-Adresse des Rechners werden auf die MAC-Adresse abgebildet.
(13) Der Haken bei "Make this virtual Machine private" wird entfernt.
(14) Die Maschinen auf einem Windows-Host laufen unter dem Account "Local System".
(15) Die Maschinen auf einem Linux-Host laufen unter dem Account "Administrator".

(1-5) Die homogene Benennung bringt Übersichtlichkeit und bietet Vorteile beim Erstellen von Scripts, da mit dem Hostnamen der virtuellen Maschine gearbeitet werden kann.

Nicht alle der vorgenannten Konventionen sind über die VMware Console einzustellen. Einige müssen durch manuelles Editieren der Konfigurationsdatei der Maschine vorgenommen werden. Dazu zählen der Name der Konfigurationsdatei und die Änderung der MAC-Adresse. Um die *vmx*-Datei umzubenennen, muss sie aus dem Inventar von VMware entfernt werden. Dazu kann die Maschine über die Konsole über den Menüpunkt "Remove from Inventory" entfernt werden. Alternativ kann auch die Befehlszeile mit dem Programm *vmware-cmd* verwendet werden. Die Befehlszeile wird vorwiegend in Batchdateien zur Automatisierung genutzt. Per Default heißt die Konfigurationsdatei immer so wie das bei der Einrichtung der Maschine ausgewählte Betriebssystem. Im oben genannten Beispiel heißt die Datei dann *Windows Server 2003 Standard Edition.vmx*. Nach dem Entfernen aus dem Inventar kann die Datei in *DCROOT01.vmx* umbenannt werden. Gleiches geschieht mit der *vmsd*-Datei, die zu diesem Zeitpunkt noch leer ist.

(6) Wird der Haken "Allocate all disk space now" gesetzt, reserviert VMware genauso viel Speicherplatz, wie die virtuelle Festplatte groß ist. Bei einer virtuellen Festplatte der Größe 36 GB würden also auch 36 GB physikalischer Speicherplatz bereits beim Erstellen des Datenträgers durch *vmdk*-Dateien reserviert. Dies bringt Performanzvorteile, allerdings belegen solche Datenträger enorm viel Speicherplatz unabhängig davon, ob sie auch mit Inhalt - also Daten innerhalb der virtuellen Maschine - gefüllt sind.
(7) Durch das Splitten der Festplattendatei in 2 GB große Stücke verhindert man Probleme mit dem Dateisystem und das Kopieren der einzelnen Festplattendateien geht schneller. So können diese Dateien auch auf eine DVD gebrannt oder auf einen mobilen Datenträger mit FAT32-Dateisystem ausgelagert werden. Werden die *vmdk*-Dateien nicht gesplittet, können Dateien in der Größenordnung von 50 oder 100 GB evtl. Probleme verursachen. Die einzelnen Festplattenfragmente werden vom System automatisch fortlaufend nummeriert, z.B.: *PC60xyz21_1-s001.vmdk*, *PC60xyz21_1-s002.vmdk*, ...
(8) Die Empfehlung von VMware lautet für Server SCSI- und für Workstations IDE-Festplatten zu verwenden. Das kann u. U. Vorteile bringen, wenn die virtuellen erstellten Maschinen zu einem späteren Zeitpunkt auf reale Maschinen übertragen werden sollen. Dieser Prozess wird als "Virtual to Physical" bezeichnet. Oft findet man dafür auch die Abkürzung V2P.
(9-10) Das Entfernen der Haben bei "Connect at power on" bringt Performanz, da das Device beim Booten nicht gesucht und nicht permanent darauf zugegriffen wird. Bei Bedarf kann das Laufwerk dynamisch zur Laufzeit der Maschine aktiviert ("gemountet") werden.
(11) Bei der Installation von VMware wird das VMware Bridge Protocol an alle Netzwerkadapter gebunden. VMware sucht sich dann automatisch einen Adapter aus, wenn in der Definition der virtuellen Maschine bei "Network Connection" der Haken bei "bridged" gesetzt wird. Um die Zuordnung eindeutig zu machen, erfolgt kein Automatismus sondern eine manuelle und feste Zuweisung der Adapter zu den virtuellen Netzwerkadaptern VMnet0 und VMnet2.
(12) Die MAC-Adresse innerhalb der *vmx*-Datei kann noch nicht geändert werden. Diese wird erst zum Zeitpunkt des ersten Starts der Maschine angelegt und verbirgt sich hinter dem Schlüssel ethernet0.generatedAddress. Auch die Schlüssel uuid.bios und die uuid.location werden zusammen mit der MAC-Adresse angelegt. Die UUID-Schlüssel entsprechen den UUID-Werten, die im BIOS eines realen Rechners gespeichert sind. Die 128 Bit langen Werte der UUID sollen weltweite Eindeutigkeit garantieren, genauso wie die MAC-Adresse der Netzwerkkarte eindeutig ist. In der Version 1 des Algorithmus zur Erstellung der eindeutigen UUID eines Rechners ist die MAC-Adresse (48 Bit) fester Bestandteil der UUID. [rfc4122]
MAC-Adressen von virtuellen Maschinen unter VMware Server besitzen folgende Struktur: 00:0C:29:xx:xx:xx. Soll der Rechner nachher beispielsweise die IP-Adresse 172.16.5.1 besitzen, wird die MAC-Adresse anhand der letzten zwei Oktette der IP-Adresse abgeändert und besitzt dann den Wert 00:0C:29:00:50:01. Dazu werden die letzten beiden Oktette der IP-Adresse bei Bedarf mit führenden Nullen aufgefüllt. Die beiden vorgenannten UUID-Schlüssel werden ebenfalls entsprechend der neuen MAC-Adresse angepasst. Diese Vorgehensweise funktioniert natürlich nur in Netzwerken vom Typ Class B und Class C. In der vorliegenden Situation ist das aber vollkommen ausreichend. In Class-A-Netzen könnten dagegen beispielsweise die letzten drei Oktette der IP-Adresse zuvor in hexadezimale Werte umgerechnet werden.
Die Anpassung der MAC-Adresse an die IP-Adresse bringt Vorteile bei der Arbeit mit einem

Netzwerkanalysator, da sich dann Pakete leichter zuordnen lassen. Im Weiteren kann dann die DHCP-Reservierung für diese virtuelle Maschine vereinfacht werden, da dort immer eine sofortige und einfache Kontrolle erfolgen kann, ob IP-Adresse und MAC-Adresse zusammenpassen. Werden IP-Adresse und MAC-Adresse gekoppelt, können garantiert keine doppelten Adressen vergeben werden. Wird die MAC-Adresse manuell verändert, müssen auch die zuvor genannten UUID-Schlüssel von Hand angepasst werden. Die automatische Anpassung der MAC-Adresse sowie der UUID-Schlüssel erledigt später das Script *vmpc.cmd* in Zusammenarbeit mit dem Perl-Script *ip2name.pl*, das mit dem Parameter "-vm" aufgerufen wird.
(13) Wird der Haken bei "Make this virtual Machine private" nicht entfernt, kann nur der Benutzer auf die Verwaltungskonsole der Maschine zugreifen, der sie erstellt hat. Durch das Entfernen des Hakens werden die Dateiberechtigungen zur Steuerung der Zugriffskontrolle benutzt.
(14-15) Unter den Optionen der Maschine kann bei "Startup/Shutdown" festgelegt werden, unter welchem Benutzeraccount die Maschinen gestartet werden. Diese Einstellung gibt es nur für Windows-Hosts und dort wird "local system" eingetragen. Dies entspricht dem lokalen Rechner. Unter Linux gibt es eine solche Einstellmöglichkeit nicht. Dort startet die virtuelle Maschine mit den Berechtigungen, die der Konfigurationsdatei (*vmx*-Datei) auf Ebene des Dateisystems zugeordnet ist. Der Besitzer der Datei wird von Root auf Administrator geändert. Der Benutzer Administrator muss dazu aber vorher angelegt und entsprechend berechtigt werden. Diese Vorgehensweise vereinfacht den Zugriff von Windows-Rechnern auf den Linux-Host und bietet unkomplizierte Möglichkeiten bei der Datensicherung des Hosts.

Prinzipielles Vorgehen bei der Erstellung der virtuellen Gastmaschinen

Von den realen Maschinen wird mit Symantec Ghost ein Image erstellt. Besitzt die reale Maschine eine feste IP-Adresse - i. d. R. nur Server - und wird daher nicht über DHCP vergeben, dann wird vor der Erstellung des Images für die aktive Netzwerkkarte eine entsprechende DHCP-Reservierung vorgenommen. Die IP-Adresse, die mit DHCP reserviert wird, entspricht dabei der festen IP-Adresse der Maschine.
Die primäre Installation eines virtuellen Gastes geschieht durch das Zurückspielen des Symantec Ghost-Images der originalen physikalischen Maschine in die virtuelle Maschine.
Dazu wird auf dem Zielsystem eine virtuelle Maschine mit den prinzipiell gleichen Parametern erstellt wie beim Echtsystem. Sind die weiter oben genannten Modifikation der virtuellen Maschine abgeschlossen und die DHCP-Reservierung vorgenommen, kann die virtuelle Maschine mit PXE ins Netz gebootet und das Image zurückgespielt werden.
In den meisten Fällen werden die virtuellen Maschinen im Anschluss an die Fertigstellung des Images nicht ordnungsgemäß starten, sondern mit einem Fehler stehen bleiben. Vielfach liegt das an dem fehlenden Treiber für das Massenspeichergerät, der einen Start der Maschine verhindert. Abhilfe schafft hier die Software VMware Converter, die in der "Starter Edition" kostenfrei erhältlich ist. Dort kann mit einem Assistenten die virtuelle Maschine für den ersten Start vorbereitet werden. Dies ist im Kapitel "Erstellung eines Windows XP Images in virtueller Umgebung" kurz und im Anhang ausführlich mit Printscreens dargestellt. Durch den Konverter wird die *vmdk*-Datei der Maschine in geeigneter Weise verändert. Der ganze Vorgang der Konvertierung dauert nur 1-2 Minuten, ist aber leider derzeit nicht zu scripten, sondern muss mittels des Assistenten ausgeführt werden.

Ist die Maschine nach dem Konvertierungsvorgang korrekt gestartet, müssen noch einige Treiber von VMware nachinstalliert werden. Ist die Maschine vor dem Image mittels *sysprep.exe* vorbereitet worden, dann ist es vorteilhaft, bereits auf der realen Maschine die Treiber für die Grafikkarte in das Verzeichnis *\PnPdrvrs* zu kopieren, so dass diese bei der Hardwareerkennung beim ersten Start der virtuellen Maschine automatisch installiert werden. Nun sollten aber noch die VMware-Tools installiert werden, die diverse Aufgaben innerhalb der virtuellen Maschine erledigen. Unter anderem sorgen sie dafür, dass die Maus- und Tastaturperformanz ernorm verbessert wird. Die VMware-Tools, die inzwischen als Open Source Code freigegeben worden sind, ermöglichen einige erweiterte Funktionen der virtuellen Gäste. So ist z.B. das automatische Herunterfahren der Gäste bei einem Shutdown des Host-Systems nur mit installierten VMware-Tools durchführbar. Die folgenden Funktionen liefern die VMware-Tools den virtuellen Gastmaschinen [Ahn07]:

- Optimierte Treiber für VGA, Maus, Netzwerkkarte und SCSI-Controller.
- Drag&Drop, Cut&Paste und Shared Folders zum komfortablen Datenaustausch zwischen Host und Gast.
- Scriptunterstützung bei bestimmten Aktionen, wie beispielsweise PowerOn, Suspend oder PowerOff.
- Bei Bedarf ist eine Zeitsynchronisation mit dem Host möglich.
- Ein Heartbeat-Signal wird regelmäßig gesendet und kann beispielsweise zur Überwachung ausgewertet werden.
- Ein Shrinken virtueller Festplatten wird möglich. Shrinken bedeutet das Verdichten und Verkleinern von Zuwachsplatten, wobei nicht mehr benötigter Speicherplatz freigegeben wird.

Die Installation der VMware-Tools erfolgt im Falle eines Terminalservers ohne die Option "Shared Folders", da dort Probleme unter Citrix bekannt sind. [VMw1317]
Die "Shared Folders" erlauben es auf einfachem Weg einen Netzwerkzugriff von Host und Gast einzurichten. Dies ist in den hier vorliegenden Fällen nicht notwendig.

Handelt es sich bei der ursprünglichen realen Maschine um einen produktiven Server, der in eine virtuelle Testanlage überführt werden sollte, ist der Transfer damit erledigt und der virtuelle Server fertig.

Handelt es sich aber um das Windows XP Basisimage aus Kapitel 7.3.1 "Erstellung des Windows XP Images" kann dieses jetzt in ein Masterimage für VMware überführt werden. Dazu werden die gleichen Schritte unternommen wie in Kapitel 7.5 "Automatisierung der Imageinstallation" erläutert. Auch wird mit der Phase 1 gestartet. Der einzige Unterschied besteht darin, dass kein Image erstellt werden muss, sondern es genügt nach Phase 1 den virtuellen Rechner herunterzufahren und die dann vorliegende Festplattendatei *PC60xyz_1.vmdk* zu kopieren und anschließend in *PC60xyz23.vmdk* umzubenennen. Ist das Splitten der Festplatten aktiv, dann sind natürlich entsprechend alle Festplatten-Fragmente umzubenennen.
So ist dann das virtuelle Pendant zum Image entstanden. Die entstandene *vmdk*-Datei ist dabei sogar noch mehr als ein Image, sie repräsentiert eine gesamte virtuelle Festplatte, im weiteren Verlauf Master-Festplatte genannt.

8.4 Automatische Softwareinstallation der virtuellen Clients

Die Phase 2 beginnt mit dem Ausbringen des Masterimages auf den Zielrechner. In der virtuellen Umgebung wird dieses Ausbringen zu einem Kopieren der Master-Festplatte auf den Zielrechner. Ist die Master-Festplatte im virtuellen Client "eingebaut" und wird dieser gestartet, läuft der Rest der Phase 2 und die darauf folgenden Phasen 3 und 4 identisch zu einem realen Client ab.
An dieser Stelle ist es relativ einfach möglich, noch zusätzliche Automatisierung mittels eines kleinen Scripts und der "virtuellen" Erweiterung des Perl-Scripts *ip2name.pl* zu erreichen. Zur Verdeutlichung des weiteren geplanten Ablaufs, dient die folgende Aufstellung in tabellarischer Form:

Schritt	Real	Virtuell
Inventarisieren der Client-Hardware inkl. MAC-Adresse	00:19:99:12:34:56	00:0C:29:12:34:56
DHCP-Reservierung anlegen	Manuelle Tätigkeit MAC: 00:19:99:12:34:56 IP: 172.16.50.90 Hostname: *PC50090* Server: *IMAGESRV*	Manuelle Tätigkeit MAC: 00:0C:29:12:34:56 IP: 172.16.60.90 Hostname: *PC60090* Server: *IMAGESRV*
Endgerät beschriften	Manuelle Tätigkeit	*vmpc.cmd* mit Parametern aufrufen
Endgerät am Arbeitsplatz aufstellen und anschließen	Manuelle Tätigkeit	*vmpc.cmd*
Endgerät starten und ggf. BIOS einstellen	Manuelle Tätigkeit	*vmpc.cmd*
Endgerät mit PXE booten und Image zurückspielen	Vorgang manuell anstoßen	*vmpc.cmd*
Endgerät installieren und betriebsfertig übergeben	Automatischer Ablauf von Phase 2, 3 und 4	Automatischer Ablauf von Phase 2, 3 und 4

Tabelle 8.3 Vergleich realer und virtueller Client-Rollout

Die obige Tabelle beschreibt den Client-Rollout, wobei ein virtueller Client prinzipiell sowohl eine Endgeräte- als auch eine Terminalserverinstallation sein kann. Im Folgenden impliziert der Namensanteil "pc" in *vmpc.cmd*, dass es sich um ein Endgerät handelt. So ist es ein leichtes eine korrespondierende *vmts.cmd* zu entwerfen oder gar mit einer universellen Batchdatei zu arbeiten, die als Parameter "PC" oder "TS" erwartet.
Die *vmpc.cmd* setzt voraus, dass

- eine DHCP-Reservierung für die gewünschte MAC-Adresse erfolgt ist,
- die MAC-Adresse als Befehlszeilenparameter übergeben wird,
- ein lesender Zugriff auf eine Freigabe existiert, wo alle "Vorlagendateien" und das Perl-Script *ip2name.pl* samt INI-Datei *ip2name.ini* liegen,
- ein schreibender Zugriff in das Zielverzeichnis möglich ist,

- VMware Server 1.x oder Workstation 5.x oder 6.x auf dem Zielrechner installiert ist.

In dem Script können bei Bedarf die folgenden Variablen geändert werden:

Variable	Default	Bedeutung
ImgSrc	*\\IMAGESRV\vmpc$*	UNC-Name der Freigabe der Quelldateien
ImgTgt	*d:\vmachine*	Zielpfad für die virtuelle Maschine
VMserver	*%ProgramFiles%\vmware\vmware server*	Pfad zur vmware.exe bei installiertem VMWare Server
VMwkst	*%ProgramFiles%\vmware\vmware workstation*	Pfad zur vmware.exe bei installierter VMWare Workstation
VMPCName	*vmpc*	Namenssuffix der virtuellen Maschine zur Anzeige in der VMware Console
VMPCvmx	*vmpc.vmx*	Zu kopierende Konfigurationsdatei
PerlPath	*%ImgSrc%\perl\bin*	Pfad zur *perl.exe*
DefaultScript	*ip2name.pl*	Name des Perl-Scripts
Parameter	-vm	Parameter für das Perl-Script

Tabelle 8.4 Variablen des Scripts *vmpc.cmd*

In der Freigabe (repräsentiert durch die Variable %ImgSrc%) liegen im Unterverzeichnis *vmpc* die folgenden Dateien *vmpc.vmx*, *vmpc.vmsd* und *vmpc_1*.vmdk*.
In dem folgenden Listing sind wieder nur die wichtigsten Stellen dargestellt und zur besseren Erläuterung nummeriert:

```
@ECHO OFF
(1) if %1.%==. GOTO MissingParameter
(2) SET MAC=%1%
(3) if exist "%VMserver%\vmware.exe" GOTO VMwareServerFound
(4) if exist "%VMwkst%\vmware.exe" GOTO VMwareWkstFound
(5) if not exist "%ImgTgt%\%VMPCName%" md "%ImgTgt%\%VMPCName%"
(6) if exist "%ImgTgt%\%VMPCName%" copy %ImgSrc%\%VMPCName%\*.*
"%ImgTgt%\%VMPCName%"
(7) if not exist %PerlPath%\perl.exe GOTO MissingPerl
(8) %PerlPath%\perl.exe %DefaultScript% %Parameter% %MAC%
    "%ImgTgt%\%VMPCname%\%VMPCvmx%"
(9) start "VMware" /D"%VM%" vmware.exe -x -c "%ImgTgt%\%VMPCname%\%VMPCvmx%"
```

(1) Wenn der Parameter MAC-Adresse fehlt, bricht das Script mit einer Hinweismeldung ab.
(2) Ansonsten wird der Wert des Befehlszeilenparameters an die Variable MAC übergeben.
(3-4) Dann wird nach einer VMware Installation gesucht und bei Erfolg an die entsprechende Sprungmarke verzweigt, ansonsten wird das Script abgebrochen.
(5) Nun wird im Pfad *%ImgTgt%\%VMPCName%* auf der lokalen Festplatte ein Verzeichnis angelegt, das den gleichen Namen trägt wie der Wert der Variable VMPCName.

(6) Dorthin werden die Dateien aus der oben genannten Freigabe kopiert.
(7) Wird kein Perl-Interpreter gefunden, bricht das Script mit einer Fehlermeldung ab.
(8) Ansonsten wird das Perl-Script mit Parametern aufgerufen.
(9) Am Ende der Batchdatei wird der neu erstellte virtuelle PC gestartet.

Nachdem der neue virtuelle Rechner gestartet ist, laufen automatisch die Phasen 2 bis 4 ab und am Ende steht ein betriebsbereiter virtueller Rechner zur Verfügung.

8.4.1 Erstellung eines Windows XP Images in virtueller Umgebung

Hier unterscheidet sich die Vorgehensweise nur wenig vor der eines realen Client. Es wird in VMware ein virtueller PC definiert, der eine vergleichbare Hardware wie ein echter PC besitzt. In unserem Fall sind die Eckdaten 1 CPU, 256-512 MB RAM, 40 GB Festplatte, CD-ROM, Sound, USB, Netzwerkkarte.

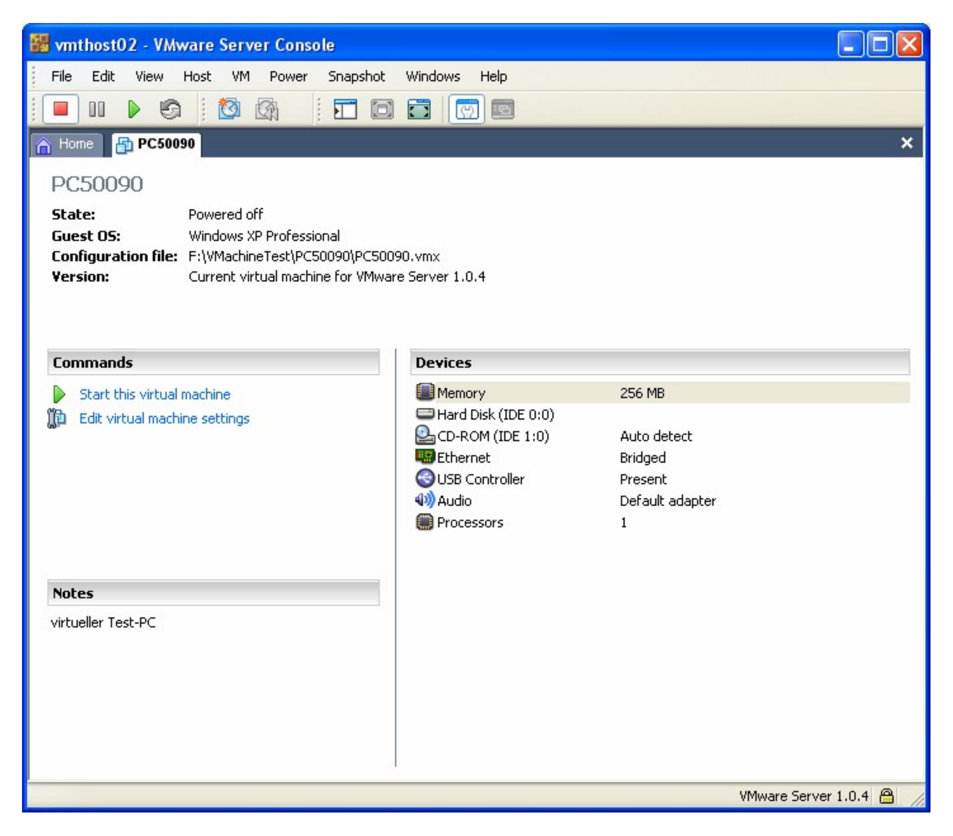


Abbildung 8.8 Konfiguration virtueller PC50090

Für die spätere Plug'n'Play-Erkennung wird analog zum Image für den Esprimo E 5915 ein Treiberverzeichnis auf *C:\PnPDrvrs\Vmware\Graphic* erstellt. In dieses wird der Inhalt des Verzeichnisses *program files\VMware\VMware Tools\Drivers\video\2k\32bit* der VMware-Tools kopiert. Die VMware-Tools, die später noch näher erläutert werden, liegen im ISO-Format[41] vor und die Datei heißt *windows.iso*. Die *windows.iso* lässt sich mit jedem beliebigen ISO-Mounter[42] öffnen und das vorgenannte Verzeichnis daraus extrahieren. Die Netzwerkkarte wird auf eine vorhandene gebridged, d.h. es wird die gleiche physikalische Verbindung des Hosts genutzt, allerdings mit einer eigenen MAC-Adresse. Die MAC-Adressen, die

---

[41] Bezeichnet ein Speicherabbild einer CD im Format ISO 9660. (International Standards Organization *http://www.iso.org*)

[42] kostenfreie Version beispielsweise Daemon-Tools *http://www.daemon-tools.cc*

VMware für die virtuellen Netzwerkkarten benutzt, beginnen alle mit der Herstellerkennung[43] 000C29. In der zugehörigen *vmx*-Datei der Maschine können wir die im Ascii-Format vorliegenden Informationen mit einem Editor ansehen. Siehe dazu den entsprechenden Ausschnitt mit der zugehörigen MAC-Adresse für unseren Beispiel-PC namens *PC50090*.

```
ethernet0.generatedAddress = "00:0c:29:38:16:06"
```

Für diese MAC-Adresse legen wir eine Reservierung im DHCP an, die sich in keinster Weise von der eines realen PCs unterscheidet. Dann kann die VMware-Maschine namens *PC50090* gestartet werden und es wird der Menüpunkt "Image zurueckspielen" ausgewählt. Man spielt aber nicht (!) das Masterimage, sondern das Basisimage *PC60xyz23.gho* zurück. Wenn wir den virtuellen *PC50090* jetzt ohne weitere Maßnahmen starten, erhalten wir einen Startfehler, einen für Microsoft typischen sogenannten Blue Screen[44] mit der Fehlermeldung STOP 0x0000007B.

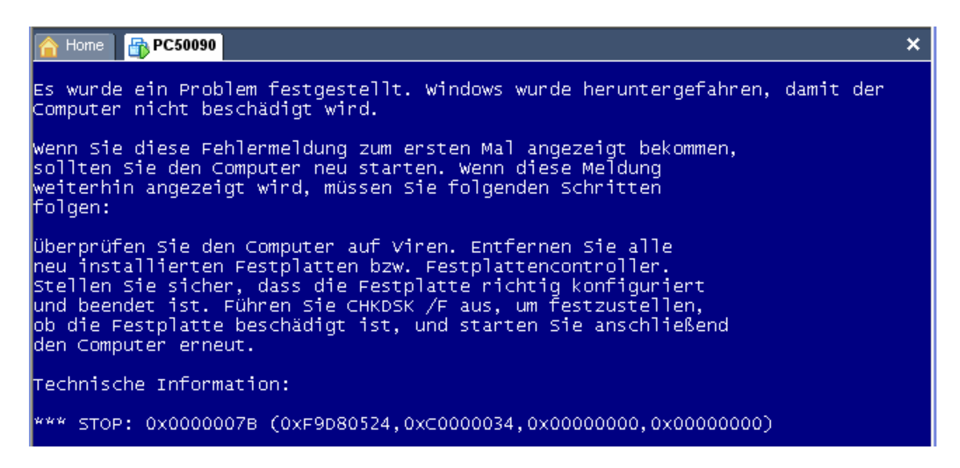


Abbildung 8.9 Blue Screen im virtuellen PC

Dieser weist auf ein Problem mit Zugriff auf den Startdatenträger hin und der Rechner bricht an dieser Stelle komplett ab und reagiert auf keine Tastenkombinationen mehr, sondern kann nur noch ausgeschaltet werden. Dieser Fehler kann auch bei realen Rechnern auftreten, es fehlt dann in den meisten Fällen ein geeigneter Treiber, um die Festplatte anzusprechen, beispielsweise beim Wechsel von einem IDE-Laufwerk auf ein SCSI-Laufwerk. Komfortable Abhilfe schafft in unserem Fall die inzwischen in einer freien Version verfügbare Software VMware Converter Starter Edition. Mit ihrer Hilfe ist es möglich, die virtuelle Maschine und den Startdatenträger anzupassen, so dass sie in einer VMware-Umgebung gestartet werden kann.

[43] Die ersten 24 Bit bezeichnen die Herstellerkennung, das entspricht 6 Zeichen in hexadezimaler Schreibweise.

[44] Im Internet auch gerne Blue Screen of Death genannt.

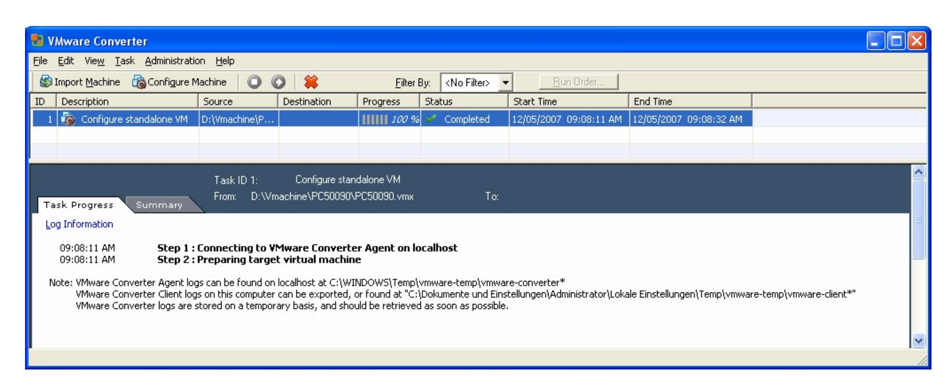


Abbildung 8.10 Ergebnis des VMware Converters

Dabei wird man mittels eines Assistenten durch den Konvertierungsprozess geführt, der am Ende den Startdatenträger, also in unserem Fall die Datei *PC50090_1.vmdk*, anpasst und für den Bootvorgang präpariert. Der gesamte Vorgang ist im Anhang mit weiteren Printscreens dargestellt. Ab hier ist die virtuelle Maschine in ihrem Bootvorgang und sonstigem Verhalten kaum noch von einer realen zu unterscheiden.

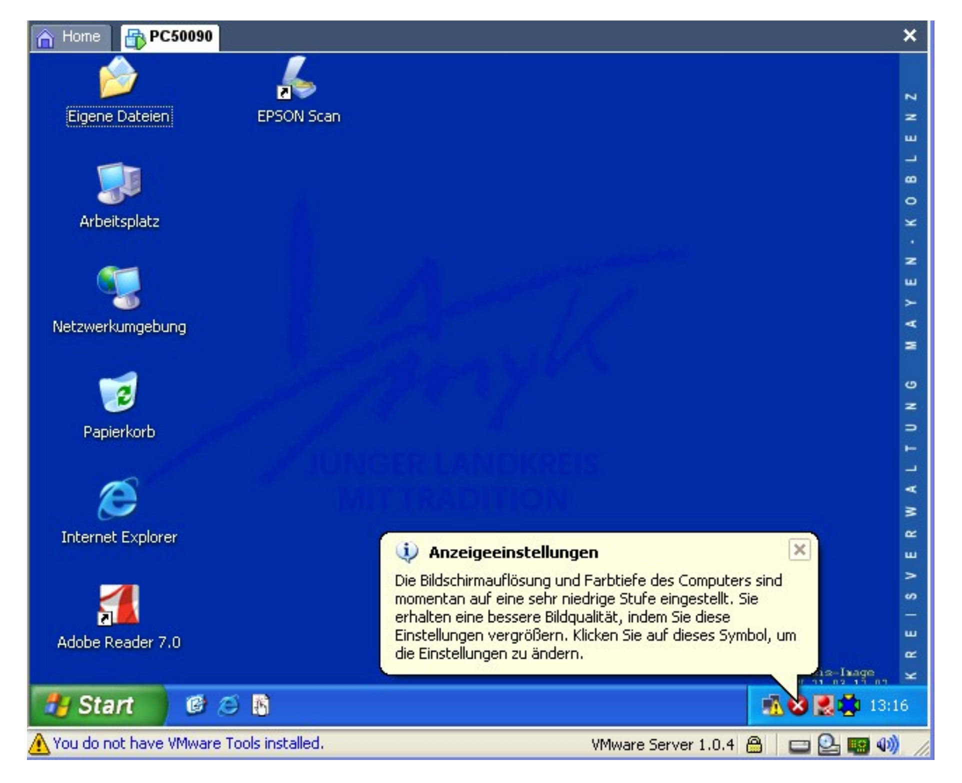


Abbildung 8.11 VMware *PC50090* ohne VMware Tools

Nach dem ersten Start müssen noch die VMware-Tools nachinstalliert werden. Diese installieren noch fehlende Treiber für virtuelle Devices nach und beschleunigen den Zugriff auf Maus und Tastatur erheblich. Erst so ist es möglich, dass Fenster der virtuellen Maschine auch ohne die Tastenkombination "Strg-Alt" zu verlassen. Die Installation wird über das Menü "VM/Install VMware Tools" der VMware Server Console gestartet und mountet dabei automatisch die Datei *windows.iso* als CD-ROM und startet über die Autorun-Funktion die Installationsassistenten. Die Installation kann nur bei laufender VMware-Maschine gestartet werden. Ein entsprechender Warnhinweis kann bestätigt werden und der Assistent kann durch Auswahl "next" gestartet werden. Die Standardvorgabe "typical" kann übernommen

werden und der Setupprozess läuft ohne weitere Abfragen innerhalb kurzer Zeit durch, wobei am Ende ein Neustart der virtuellen Maschine verlangt wird.
Nach dem Neustart verrät ein Blick auf den Gerätemanager, dass alle Devices mit den korrekten Treibern ausgestattet sind. Auffällig dabei ist der SCSI-Treiber, der bei VMware Maschinen immer zu finden ist, da er in einer assistentengestützten Definition nicht weggelassen werden kann.

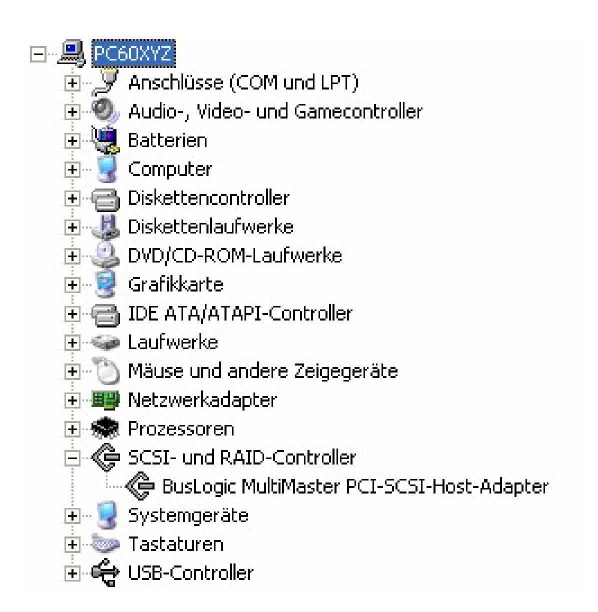


Abbildung 8.12 Gerätemanager nach der Installation der VMware Tools

Nun kann die Maschine, analog zu den realen Maschinen, durch den automatischen Login auf den *Sysprep*-Vorgang vorbereitet werden. Dann wird die Maschine heruntergefahren und die Festplattendatei der Maschine, in dem Beispiel die Datei *PC50090_1.vmdk*, kann weg kopiert werden. Ein erneutes Erstellen eines Masterimages mit Ghost entfällt, da man hier den Vorteil hat, dass in jeder neuen Maschine, die erstellt werden soll, einfach die vorhandene virtuelle Festplatte in Form der Datei *PC50090_1.vmdk* eingebunden werden kann.

An dieser Stelle wird deutlich, dass die Definition der virtuellen Maschine und deren Name *PC50090* sowie die dazugehörigen physikalischen Dateien (**.vmx, *.vmdk, *.vmxf, *.vmsd*), nichts mit der Installation innerhalb dieser virtuellen Hardware zu tun haben. Die virtuelle Hardwaremaschine heißt *PC50090*, die Workstationinstallation dieser Maschine benutzt den Hostnamen *PC60xyz*. Um bei der Analogie zu den realen Images und in der korrekten Namenssyntax zu bleiben, wird das "VMware-Image" von *PC50090_1.vmdk* in *PC60xyz23_sysprep_1.vmdk* umbenannt.

Soll nun ein neuer virtueller Client installiert werden, so wird in folgender Reihenfolge vorgegangen:

1. virtuelle Hardware mit Betriebssystem Windows XP Professional definieren (mit Assistent)
2. Festplatte in virtueller Hardware entfernen
3. virtuelle Maschine booten und wieder anhalten
4. Reservierung für virtuellen PC anlegen
5. virtuellen PC booten, Reservierung testen, PC danach anhalten
6. Festplattendatei *PC60xyz23_sysprep_1.vmdk* in das Verzeichnis des virtuellen PCs kopieren
7. *PC60xyz23_sysprep_1.vmdk* in *Hostname_1.vmdk* umbenennen

8. virtuelle Festplatte *Hostname_1.vmdk* hinzufügen (mit Assistent)
9. virtuellen PC starten

(1) Es ist selbst im angepassten Modus (Virtual Machine Configuration = Custom) nicht möglich, mit dem Assistenten von VMware eine Maschine ohne Festplatte zu definieren.
(2) Danach muss die Festplatte wieder aus der Konfiguration gelöscht werden.
(3) Erst beim Start des virtuellen PCs wird die MAC-Adresse in der Konfiguration erzeugt.
(4) Nun kann anschließend die Reservierung im DHCP-Server angelegt werden.
(5) Diese Reservierung sollte einmal auf korrekte Funktion getestet werden.
(6) Das Image in Form der Festplattendatei *PC60xyz23_sysprep_1.vmdk* wird in das dem virtuellen PC zugehörige Verzeichnis kopiert.
(7) Dort wird sie entsprechend dem Hostnamen des neuen PCs umbenannt.
(8) Mit dem Assistenten für die Hardware wird dem virtuellen PC die umbenannte Datei als Festplatte hinzugefügt. Dazu wird als Typ der Festplatte mit dem Assistenten "use an existing virtual disk" ausgewählt.
(9) Nun ist alles bereit für den Start des virtuellen PCs und wir befinden uns nun am gleichen Punkt wie bei einem realen PC nach dem ersten Start nach dem Zurückspielen des Images *PC60xyz23_sysprep.gho*.

8.4.2 Erstellung eines eLux NG Images in virtueller Umgebung

Mit der Einführung der neuen Endgeräte vom Typ Esprimo E 5915 wurde ein Update der Betriebssystems auf Version 1.40 oder höher notwendig. Die derzeit aktuelle Version ist 1.45-3. Diese Version kann auch unter VMware mit dem Intel-Container ohne Einschränkungen verwendet werden. Dies war in der Version 1.37-3 nicht der Fall, dort musste unter VMware zwingend der PC-Container genutzt werden.
Die Installation der VMware-Tools wird derzeit nicht unterstützt. Die Performanz der Endgeräte ist aber zu Testzwecken, also in der Testanlage oder am Arbeitsplatz eines Administrators vollkommen ausreichend. Bei der Installation der Maschine wird als Betriebssystem Linux und dann "Other Linux 2.4.x Kernel" ausgewählt.

Im Falle von eLux NG kann die Installation analog zu realen Maschinen vorgenommen werden, d.h. die DHCP-Reservierung des virtuellen Endgerätes wird erstellt und dann der Rechner über PXE ins Netz gebootet und der Rechner mit dem Menüpunkt "A" installiert. Die Installation dauert nur ca. 60 Sekunden länger als bei einem realen Client und liegt mit einer Gesamtzeit von 150 Sekunden immer noch sehr niedrig.

8.4.3 Erstellung eines Server Images in virtueller Umgebung

Die Erstellung unterscheidet sich kaum von Windows XP. Das Betriebssystem ist natürlich Windows Server 2003 und bei der Installation der VMware-Tools werden die "Shared Folders" nicht installiert. Die Definition der virtuellen Hardware ist an die Anforderungen eines Servers angelehnt. Sind mehr als eine Festplatte vorhanden, werden diese entsprechend der vereinbarten Namenskonventionen angelegt. RAID1- oder RAID5-Verbände machen in der virtuellen Maschine wenig Sinn, da die Host-Maschine die Ausfallsicherheit von RAID-Verbänden zur Verfügung stellt. Soll die Maschine mehrere virtuelle CPUs besitzen, kann dies ebenfalls bei der Definition des virtuellen Servers berücksichtigt werden, da bis zu zwei CPUs einzustellen sind. Dabei muss natürlich der Host auch mindestens 2 CPUs besitzen.
Ansonsten ist die Vorgehensweise identisch zu Kap. 8.4.1 "Erstellung eines Windows XP

Images in virtueller Umgebung" und in den vorherigen Kapiteln bereits entsprechend erläutert.

Für den Fall, dass eine reale Maschine in die Testanlage in eine virtuelle Umgebung transferiert werden soll, kann es vorkommen, dass große Datenmengen übertragen werden müssen. Dies ist beispielsweise bei Datenbank-, File- und Mail-Server sehr wahrscheinlich. Dort ist es angebracht, von dem Betriebssystem ein Image zu erstellen und die Daten des Servers separat mittels einer Datensicherung in die virtuelle Umgebung zu übertragen. Dies erfordert allerdings, dass die Installation des Betriebssystems inkl. Programme von den Daten getrennt auf verschiedenen Partitionen oder Festplatten aufbewahrt wird. Die Sicherung kann mit dem betriebssystemeigenen Programm "ntbackup" erledigt werden. Die vorgenannte Vorgehensweise entspricht in etwa dem Vorgehen im Falle eines "Desaster Recovery", wie er auch in der realen produktiven Umgebung vorkommen kann. Kann ein Server aufgrund eines Fehlers nicht mit einer einfachen Fehlerbehebung oder Datenrücksicherung wiederhergestellt werden, spricht man von einem dann notwendigen "Desaster Recovery". Dies entspricht der Wiederherstellung eines Servers, auf dem sich, beispielsweise nach einem Hardwaredefekt, keinerlei Daten mehr auf den Festplatten befinden - weder Betriebssystem noch Programmdaten. Dort wird i. d. R. ein vorhandenes Image des Servers zurückgespielt und anschließend werden die aktuellen Daten aus der vorhandenen täglichen Datensicherung zurückgespielt.

8.5 Zusammenfassung der Installationsmethoden

Aus den letzten Kapiteln lassen sich die beiden folgenden Vorgehensweisen für die Verteilung von Betriebssystemen im virtuellen Umfeld festlegen:

1. Prinzipiell kann die Softwareverteilung mit virtuellen Maschinen analog zu den realen Geräten erfolgen.
2. Es ist möglich, die Verteilung über Symantec Ghost durch die Verteilung bzw. das Kopieren von virtuellen Festplatten zu ersetzen.

Sind die virtuellen Computer einmal definiert, also ihre virtuelle Hardware erstellt, können die gleichen Techniken wie bei realen Maschinen benutzt werden. Allerdings werden bei virtuellen Computern beim PXE-Start und anschließenden Image-Vorgang mit Ghost, Paket-Treiber von Symantec benutzt, die nur eine geringe Performanz bieten. Der Paket-Treiber *Undipd.com* von Symantec hat unter VMware nur ca. 15-20% der Performanz wie vergleichbare reale Maschinen in einem 100 MBit Netzwerk. Daher empfiehlt es sich, den Image-Anteil beim Transfer von realen in virtuelle Maschinen so gering wie möglich zu halten. Es bietet sich daher die im vorherigen Kapitel angegebene Vorgehensweise in Verbindung mit einer Datenrücksicherung über ein Backupprogramm an. Ab der Version 2 des VMware Servers, der aktuell in der Betaversion verfügbar ist, wird das "ghost boot package" unterstützt. Dort allerdings "nur " in Verbindung mit der Installation von Windows 2000. Es ist aber damit zu rechnen, dass auch Windows XP und Server 2003 entsprechend performant sein werden, wenn auch nur offiziell mithilfe der Windows Server 2003 Automated Deployment Services.

Werden neue virtuelle Maschinen – Endgeräte oder Terminalserver – ausgerollt, ist die Nutzung von *vmpc.cmd* die optimale Vorgehensweise beim Deployment. Die Geschwindigkeit hängt dort fast ausschließlich von der Größe der zu übertragenden Dateien, der vorhandenen Netzwerkgeschwindigkeit und der Performanz des lokalen Datenträgers des Ziel-

Hosts ab.
Wie bereits in der Tabelle 8.3 "Vergleich realer und virtueller Client-Rollout" aufgeführt, ist es mithilfe von *vmpc.cmd* möglich, den kompletten Rollout von virtuellen Maschinen zu automatisieren. Auch in der bereits angesprochen Erweiterung zur universellen Nutzung des vorgenannten Scripts in Verbindung mit einer automatischen Installation einer kostenfreien VMware Server Umgebung, kann der gesamte Vorgang auf einem Client automatisiert werden, der bis dahin noch keine Virtualisierungssoftware besitzt.

9 Ergebnis

Die Entscheidung, ob ein Endgerät ein Thin- oder Fat-Client wird, kann an einer einzigen zentralen Stelle getroffen werden. Im DHCP wird als Ziel für den Start über PXE entweder der Server *SCOUTNG* oder *IMAGESRV* festgelegt. Das Thin-Client Betriebssystem eLux NG läuft mit dem einzigen Image der Version 1.45-3 auf Endgeräten vom Typ

- FSC Esprimo E 5700,
- FSC Esprimo E 5915 und
- VMware.

Ein Einsatz auf einem Laptop ergibt in diesem Zusammenhang keinen Sinn, da diese Geräte vorwiegend für den mobilen Einsatz gebraucht werden. Denkbar wäre allerdings ein Einsatz von Laptops als platzsparende Heimarbeitsplätze, die mittels eines entsprechenden Clients eine gesicherte VPN-Verbindung über das Internet zur KVMYK aufbauen.
Ein einziges Masterimage - in der Version 23 - ist auch die Grundlage aller im Hause eingesetzten Rechner vom Typ

- FSC Esprimo E 5700,
- FSC Esprimo E 5915,
- FSC Scenic L D1386,
- FSC Scenic P 300 D1561,
- FSC Scenic E 600 D1534,
- FSC Lifebook E 8310,
- FSC Celsius H 250 und
- VMware.

Die Pflege von DHCP und Images reduziert sich hier jeweils auf eine einzige Stelle. Im Falle von VMware ist die prinzipielle Möglichkeit gegeben, das Ghost-Image zu benutzen. Wesentlich effektiver ist der Einsatz eines konvertierten Images, das innerhalb kurzer Zeit aus dem Ghost-Image gewonnen werden kann. (VMware Converter)
Ein Wechsel von einem Thin- auf einen Fat-Client erfordert nur eine Änderung im DHCP, anschließend einen Neustart und das Einleiten eines PXE-Boot. Die Umstellung ist in beide Richtungen möglich und jeweils in wenigen Minuten abgeschlossen. Die Umstellung ist so einfach, dass sie nach Absprache mit dem IT-Support auch von einem Endanwender durchgeführt werden kann. Eine Performanzmessung hat in einem 100 MBit Netzwerk, inkl. Hardwareerkennung bis zur betriebsbereiten Anmeldung durch den Anwender, eine Dauer von ca. 13 Minuten ergeben. Unter optimalen Bedingungen sind wesentlich bessere Ergebnisse möglich.
Die Namensgebung der zu installierenden Geräte geschieht vollautomatisiert und kann in Abhängigkeit der IP-Adresse (`mode = ip`) oder MAC-Adresse (`mode = http|local`) erfolgen und lässt damit alle Möglichkeiten der Namensgebung zu. (*ip2name.cmd* u. *ip2name.pl*)

Für Endgeräte wird im Anschluss an die Systemvorbereitung ein Dienst installiert, der es ermöglicht auf den Geräten nachträglich Software zu installieren. (PCinst)

Analoges wurde für die Installation der Terminalserver erreicht: Ein einziges Image, das die Server nach der Installation automatisch zum Active Directory und der Citrix Farm hinzufügt (*1login.cmd*). Die Installation der Anwendungen geschieht hier automatisch durch den Citrix Installationmanager.

Die Fortführung und Erweiterung der Images ist grundsätzlich problemlos realisierbar, sowohl für zukünftige neue Hardware, als auch für neue Software und Programmupdates. Aus Erfahrung ist allerdings damit zu rechnen, dass früher oder später Inkompatibilitäten zwischen Treibern oder Softwareinstallation auftreten können, die gelöst werden müssen. Trotz der umfangreichen Software- und Treiberinstallationen ist das Image der Endgeräte derzeit mit einer Größe von nur 1,5 GB sehr klein und daher entsprechend schnell verteilt. Das Server-Image ist nur 0,9 GB klein.

Beide Images, sowohl das der Endgeräte als auch das der Terminalserver, können in einer VMware-Umgebung eingesetzt werden. Dazu wurde der besondere Komfort geschaffen, dass der gesamt "Rollout" inkl. Installation bis hin zum Start der virtuellen Maschinen automatisch abläuft. (*vmpc.cmd*)

In einer VMware funktioniert kein Wake On LAN. Dies ist evtl. aber in der Version Workstation 6.0 möglich, zumindest existiert in der *vmx*-Datei ein vielversprechender Eintrag "ethernet0.wakeOnPcktRcv = "FALSE". Allerdings ist derzeit keine Dokumentation zu der Funktionsweise des Parameters erhältlich. Die Wake On LAN-Funktionalität ist nur im Falle eines Thin-Clients derzeit sinnvoll im Einsatz. Da aber virtuelle Thin-Clients derzeit nur zum Test und am Arbeitsplatz eines Administrators eingesetzt werden, ist die fehlende Funktionalität keine große Beeinträchtigung.

Vollautomatisierung

Eine Vollautomatisierung ist im Bereich von eLux NG und Scout NG ab der Version 9.7 der Verwaltungskonsole möglich. Dies wurde von dem Hersteller in einer Mail vom 14.12.2007 bestätigt und die Version 9.7 ist am 22.02.2008 erschienen. Dann entfällt das Drücken der Taste "A" und die entsprechende Auswahl ist als Kernel-Parameter mit dem Aufruf fest zu hinterlegen.

Eine vergleichbare Vollautomatisierung ist auch im Bereich der Fat-Clients möglich, aber unter Sicherheits- und Performanzgesichtspunkten derzeit nicht realisiert. Dazu muss

1. die Bootreihenfolge auf primäres Bootmedium "PXE" und sekundäres Medium "Festplatte" getauscht werden,
2. an zentraler Stelle (DHCP) die Bootoption 66 "Name der Startdatei" bei Bedarf geändert werden,
3. der Ghostcast Server automatisch gestartet werden.

Dazu wird auf dem *IMAGESRV* die Datei *pxelinux.0* nach *pxelinux.1* kopiert und dann die *pxelinux.0* so geändert, dass als einziger Default-Eintrag "von Festplatte starten" bestehen bleibt. Auf dem Server *SCOUTNG* wird die vorhandene *pxelinux.0* in *pxelinux.1* umbenannt und die *pxelinux.0* vom *IMAGESRV* auf den *SCOUTNG* kopiert. In der Folge starten dann alle Endgeräte als erstes Medium über das Netzwerk, laden dort die *pxelinux.0*, warten 5 Sekunden und starten dann weiter von der Festplatte.

In der Datei *pxelinux.1* auf dem *IMAGESRV* wird dann als einziger Default-Eintrag "Image zurueckspielen" bzw. beim *SCOUTNG* die automatische Installation von eLux NG ausge-

wählt. Wird die *pxelinux.1* gestartet, dann wird entweder ein Ghost-Image auf das Endgerät gespielt oder das Endgerät wird mit eLux NG konfiguriert. Auf dem *IMAGESRV* wird die *pxelinux.1* nach *pxelinux.2* kopiert und dort als einziger Default-Eintrag "Image erstellen" hinterlegt. Analog werden die BIOS-Updates hinterlegt, so dass man die folgende Auswahl bekommt:

Aktion	Datei auf IMAGESRV	Datei auf SCOUTNG
von Festplatte starten	*pxelinux.0*	*pxelinux.0*
Image zurueckspielen	*pxelinux.1*	-
eLux NG installieren	-	*pxelinux.1*
Image erstellen	*pxelinux.2*	-
GHOST Walker starten (nach Image zurueckspielen)	*pxelinux.3*	*pxelinux.3*
Ghost manuell starten	*pxelinux.4*	*pxelinux.4*
BIOS-Update und SET E-600 (D1534)	*pxelinux.5*	*pxelinux.5*
BIOS-Update und SET Esprimo E5700 (D2334)	*pxelinux.6*	*pxelinux.6*
BIOS-Update und SET Scenic P-300 (D1561)	*pxelinux.7*	*pxelinux.7*
BIOS-Update und SET Scenic L (D1386)	*pxelinux.8*	*pxelinux.8*

Tabelle 9.1 TFTP-Bootdateien bei Vollautomatisierung

Die Unterscheidung, was das Endgerät starten soll, wird dann im DHCP getroffen, indem der Default-Eintrag der Startdatei (Option 66) *pxelinux.0* durch eine der oben genannten Einträge überschrieben wird.

Der Ghostcast Server ist grundsätzlich über Befehlszeilenschalter zu automatisieren. Die Kontrolle des Vorgangs "Image erstellen" oder "Image zurueckspielen" wird schwieriger und aufwändiger zu überwachen. Zudem bringt das Ändern der Bootreihenfolge im BIOS des Rechners bei jedem Start eine unnötige Verzögerung mit. Die TFTP-Server müssen darüber hinaus redundant ausgelegt und jederzeit verfügbar sein, da ansonsten die Gefahr besteht, dass die Endgeräte bei Start den TFTP-Server nicht finden und hängen bleiben.

Unter dem Aspekt, dass das Imagen, außer beim erstmaligen Ausbringen, ein partieller Vorgang ist, macht das Vollautomatisieren in dem vorliegenden Fall keinen Sinn, obgleich es grundsätzlich im Bereich des Möglichen ist.

10 Zusammenfassung

Im Rahmen der Migration von MS Windows NT zu Windows XP bzw. Windows Server 2003 erfolgte die Neubeschaffung von 18 Servern. Der Austausch der Endgeräte wurde zweistufig vorgesehen. In der Stufe 1 wurden ca. 370 neue und ca. 25 bestehende Geräte neu installiert, in der nächsten Stufe folgen noch ca. 25 Geräte. Inzwischen wurden 14 reale und virtuelle Server und ca. 15 virtuelle Endgeräte automatisch installiert. Weiterhin wurden ca. 15 reale Server in virtuelle überführt.

Standardisierung

Neu geschaffene Namenskonventionen und feste Zuordnung von Gerätenamen mit zugehörigen quasistatischen IP-Adressen sorgen für klare und einfache Verhältnisse. Dies trifft sowohl in der Verwaltung von DHCP als auch im Active Directory und in der Inventardatenbank deutlich zutage. Ein Wechsel des Namens bei Umzug oder Neuinstallation ist nicht mehr notwendig. Durch lange Garantiezeiten von fünf Jahren kann eine stabile Umgebung mit wenig Änderung in der Hardware für einen langen Zeitraum vorhergesagt werden. Die Definition eines Standardarbeitsplatzes inkl. Standardsoftware sorgt für eine einheitliche und homogene Softwareumgebung. Unterstützt wird das durch die Definition eines Warenkorbs, der für klare und standardisierte Verhältnisse sorgt.

Homogenisierung

Die Reduzierung der Anzahl der Images pro Betriebssystem – Windows XP, eLux NG und Windows Server 2003 – auf ein einziges ist das Optimum. Gleiches gilt für die virtuelle Umgebung: Ein einziges Image pro Betriebssystem. Dieses Ziel konnte erreicht werden. Durch die Reduzierung der Fülle der eingesetzten Hardware, u.a. durch Bildung eines Warenkorbs, kann die Treibervielfalt gering gehalten werden. Die konsequente Fortführung der Terminalservertechnik führte zu einem Anteil von derzeit 78% Thin-Clients. Nutzung gleicher Installationsquellen und gleicher Software für Terminalserver und Endgeräte sorgen für einheitliche Bedingungen in beiden Einsatzfeldern. Fehlersituationen, die auf unterschiedliche Software zurückzuführen sind, sind bis auf wenige Ausnahmen ausgeschlossen. Eine ausführliche Dokumentation der Images sorgt für klare und nachvollziehbare Verhältnisse.

Automatisierung

Bis auf wenige Ausnahmen, wie das Auspacken, Inventarisieren und Aufstellen der Hardware, ist der weitere Prozess der Installation der Arbeitsplätze komplett automatisiert. Dies gilt sowohl für Endgeräte als auch für Terminalserver. Somit kann die Gleichheit aller Installationen gewährleistet werden. Dies wurde einerseits erreicht durch das konsequente Ausnutzen der Möglichkeiten des Betriebssystems bzw. seiner mitgelieferten Tools, wie z.B. der Systemvorbereitungsdienste von Microsoft. In Verbindung mit Symantec Ghost und dem

Schreiben eines Programms zur automatischen Namensgenerierung und Namenszuweisung (*ip2name.pl*) andererseits, wurde es erst möglich den Installationsprozess voll zu automatisieren. Durch das geschickte Zusammenspiel von selbst erstellten Scripts (*ip2name.cmd* etc.), ist es möglich den Benutzereingriff zu minimieren und trotzdem eine schnelle, sichere und einfache Installation zu gewährleisten. Im Anschluss daran sorgen die Gruppenrichtlinien des Active Directory dafür, dass die Endgeräte und Server die Einstellungen erhalten, die ihnen zugewiesen sind. Diese Einstellungen erhalten sie einerseits durch Mitgliedschaft in einer bestimmten Gruppe und andererseits durch die Position, die sie als Objekt im Active Directory einnehmen. Beides wird durch den Installationsprozess mittels Scripts (*dsmodpc.cmd* bzw. *dsmodts.cmd*) und Einträgen in der *sysprep.inf* erreicht. Im Rahmen der Erstinstallation werden ebenfalls automatisch zusätzliche Dienste installiert. Für das Endgerät ist das der Dienst "PCinst" und für die Server ist dies die Installation von Citrix Presentation Server, die die Citrix Terminaldienste zur Verfügung stellen.

Flexibilisierung

Durch die Schaffung einer universellen Hardwareplattform (E 5700 und E 5915) für Thin- und Fat-Client ist es möglich, ein- und denselben Arbeitsplatz als Thin- oder Fat-Client zu betreiben. Aufwändiger Hardwareaustausch und Umbau von Geräten kann künftig entfallen. Der Schulungsraum mit 9 Geräten kann so innerhalb von kürzester Zeit wahlweise mit Thin-Clients oder Fat-Clients ausgerüstet werden. An zentraler Stelle im DHCP kann so entschieden werden, ab ein Endgerät als Thin- oder Fat-Client installiert wird. Durch die Konvertierung des Images in die virtuelle Umgebung, kann auch dies beliebig mittels VMware gestartet werden. Die Software bzw. die Installation von eLux NG läuft sowohl real als auch virtuell gleich ab und ist somit vollkommen flexibel einsetzbar. Die Vorbereitung eines virtuellen Terminalserver-Images ermöglicht jederzeit eine schnelle Bereitstellung eines virtuellen Servers, um beispielsweise Softwareevaluierungen durchzuführen.

Virtualisierung

Die konsequente Fortsetzung der erarbeiteten Konzepte der automatischen Installation ist theoretisch 1:1 in die virtuelle Umgebung umsetzbar. Einzig der Schritt der Konvertierung des Images mittels des VMware Converters ist derzeit noch nicht komplett zu automatisieren. Eine bessere Variante bietet die Möglichkeit, anstatt Images entsprechend vorbereitete komplette virtuelle Festplatten zu verteilen. Diese liegen schließlich in Form von einer oder mehreren Dateien vor, die kopiert werden können. Die optimierte Lösung bietet die Möglichkeit, quasi einen kompletten Rechner zu verteilen (*vmpc.cmd*). Dies kann theoretisch sowohl für virtuelle Endgeräte als auch Server vorgenommen werden.
In einem weiteren Schritt erfolgte die Installation der Testanlage. Die Umsetzung erfolgte auf Basis zweier vorhandener Server, die geringfügig erweitert wurden. Als Betriebssysteme wurde auf den Hosts Linux und Windows eingesetzt. Die Virtualisierung erfolgte mit der kostenlosen Software VMware Server. Vorhandene produktive Server wurden per Image und anschließendem Konvertierungsvorgang in die Testanlage übertragen. Die Sicherung der virtuellen Server geschieht auf den Hostmaschinen, die abends alle virtuellen Server über Kreuz auf separate Festplatten kopieren.

Die Auswirkungen der oben genannten Maßnahmen haben für einen spürbaren Rückgang des Supportaufkommens gesorgt. Wenn Fehler oder Probleme auftreten, dann sind es i. d. R.

prinzipielle Probleme oder Fehler in den Versionen der Software der Hersteller, die in bisherigen Tests noch nicht aufgefallen sind. Einzelprobleme dagegen treten kaum noch auf.

Die durchgeführten Serverinstallationen sind grundsätzlich auch auf andere Produkte übertragbar; z.B. Mail-Server, Domänencontroller oder SQL-Server könnten damit nach geringfügiger Modifikation automatisch installiert werden. Das vorgestellte Konzept funktioniert durchweg auf allen MS-Betriebssystemen, die *sysprep.exe* unterstützen und in einem TCP/IP basierten Netzwerk betrieben werden. Da auch die neuen imagebasierten Betriebssysteminstallationen -wie beispielsweise Vista - noch die Systemvorbereitungsdienste unterstützen, kann dieser Vorgang auch für diese Betriebssysteme nahezu unverändert genutzt werden. Alle genutzten Schlüssel aus der *sysprep.inf* von Windows XP werden in Vista noch unterstützt.

11 Ausblick

Da in naher Zukunft noch weitere PCs vom Typ FSC Esprimo E 5915 ausgerollt werden, muss noch das automatische Einstellen des BIOS über PXE angepasst werden. Gleiches gilt für eine größere Anzahl Laptops vom Typ FSC Lifebook E 8310. Im Bereich der Server muss das Serverimage noch auf eine etwas ältere Plattform vom Typ HP Proliant DL 380 G3 übertragen werden.

Eine sicherlich reizvolle Aufgabe ist die Übertragung des vorliegenden Konzepts auf Windows Vista bzw. den in der Beta-Version vorliegenden Windows Server 2008. Beide liegen erstmals in einem windowseigenen Image-Format vor und bringen eigene kostenfreie Mittel zur Erstellung und Verteilung von Images mit. Für Windows Vista nennt sich das Produkt WAIK (Windows Automated Installation Kit). Im Zusammenspiel mit dem Serverdienst "Windows Deployment Services" lassen sich damit Vista und Server 2003 komfortabel verteilen. Damit ist zwar möglich, Windows XP zu verteilen, allerdings nur über den Umweg mittels *sysprep.exe*. Auch hier ist die automatische Namensgebung nur unbefriedigend gelöst.

Weitere wünschenswerte Eigenschaften des Programms *ip2name.pl* sind in der nach folgenden Liste aufgezählt. Deren Realisierung hängt u. a. von der weiteren Entwicklung der Betriebssysteme und den Anforderungen ab, die sich daraus ergeben.

- Unterstützung von FTP zum Download der *ip2name.txt*;
- Datenbankunterstützung (ODBC, SQL), d.h. Abfrage des Namens aus einer DB;
- Betriebssystemunabhängigkeit:, d.h. Übertragung des Konzepts auf Linux-Derivate;
- Weitere Administrative Tasks, z.B. eine automatische Berichtserstellung für die Inventardatenbank;
- Ausführliche Protokollierung in eine Logdatei;
- E-Mail-Unterstützung, z.B. Benachrichtigung nach Abschluss eines Images;
- Sicherheitsstufe erhöhen, z.B. durch Umwandlung von Batchdateien in ausführbare *exe*-Dateien oder durch Verschlüsselung von Kennwörtern;
- Auslesen der DHCP-Option 12 (Hostname), um diesen als Computernamen zu verwenden;

Images, die mit Norton Ghost Version 9.0, 10.0, und 12.0 (mit der Dateiendung *sv2i*) oder Acronis True Image 9 erstellt worden sind, können direkt mit dem VMware Converter bearbeitet werden, d.h. hier entfällt der Schritt, das Ghost-Image erst in eine virtuelle Maschine einzuspielen. Unter diesem Aspekt lohnt sich eine Überprüfung, ob ein Update sinnvoll ist.

Da sich das vorliegende Konzept bewährt hat und die Virtualisierung mit ihren enormen Vorteilen weiter voran schreitet, ist davon auszugehen, dass sich die erarbeitete Vorgehensweise in Zukunft bezahlt macht und mit den oben genannten Möglichkeiten sogar langfristig vereinfacht.

12 Anhang

12.1 Softwarequellen und Links

12.1.1 Verwendete Programme

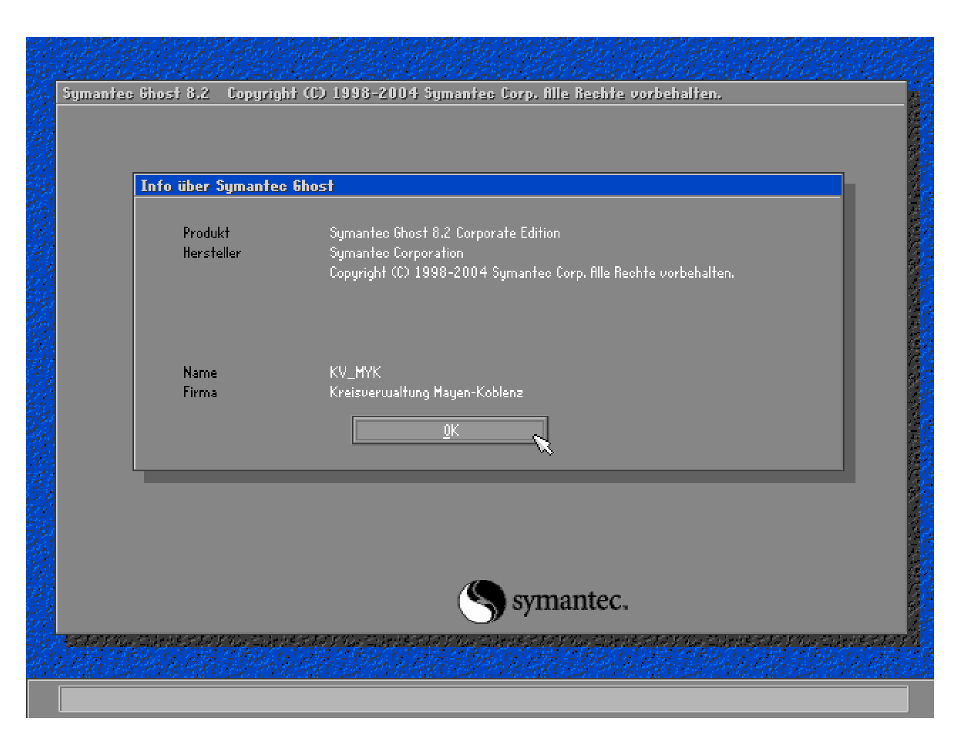
Activestate Perl for Windows	http://www.activestate.com/Products/activeperl/
DAEMON Tools Lite	http://www.daemon-tools.cc
Microsoft Sysinternals Tools	http://www.microsoft.com/technet/sysinternals/
Microsoft Sysprep (Deployment-Tools) für Windows XP SP2	http://www.microsoft.com/downloads/details.aspx?displaylang=de&FamilyID=3e90dc91-ac56-4665-949b-beda3080e0f6
Microsoft Sysprep (Deployment-Tools) für Windows Server 2003	http://www.microsoft.com/downloads/details.aspx?FamilyID=93F20BB1-97AA-4356-8B43-9584B7E72556&displaylang=de
Microsoft Windows Server 2003 ResourceKit Tools	http://www.microsoft.com/downloads/details.aspx?FamilyID=9D467A69-57FF-4AE7-96EE-B18C4790CFFD&displaylang=en
NetSupport Manager - Remote Control	http://www.netsupportmanager.com/DE/index.asp
Novell SuSE OpenSuSE 10.3	http://www.opensuse.org
PuTTY, a free Telnet/SSH-Client	http://www.chiark.greenend.org.uk/~sgtatham/putty/
Rawrite für Windows	http://www.chrysocome.net/rawwrite
redWall, a bootable CD-ROM Firewall	http://www.redwall-firewall.com
Symantec Ghost	http://www.symantec.com/de/de/enterprise/products/overview.jsp?pcid=1025&pvid=865_1
Unicon Software GmbH; Elux NG und Scout NG	http://www.myelux.de/
VMware Server	http://www.vmware.com/de/products/server/
VMware Converter Starter Edition	http://www.vmware.com/de/products/converter/

12.1.2 Alternativen

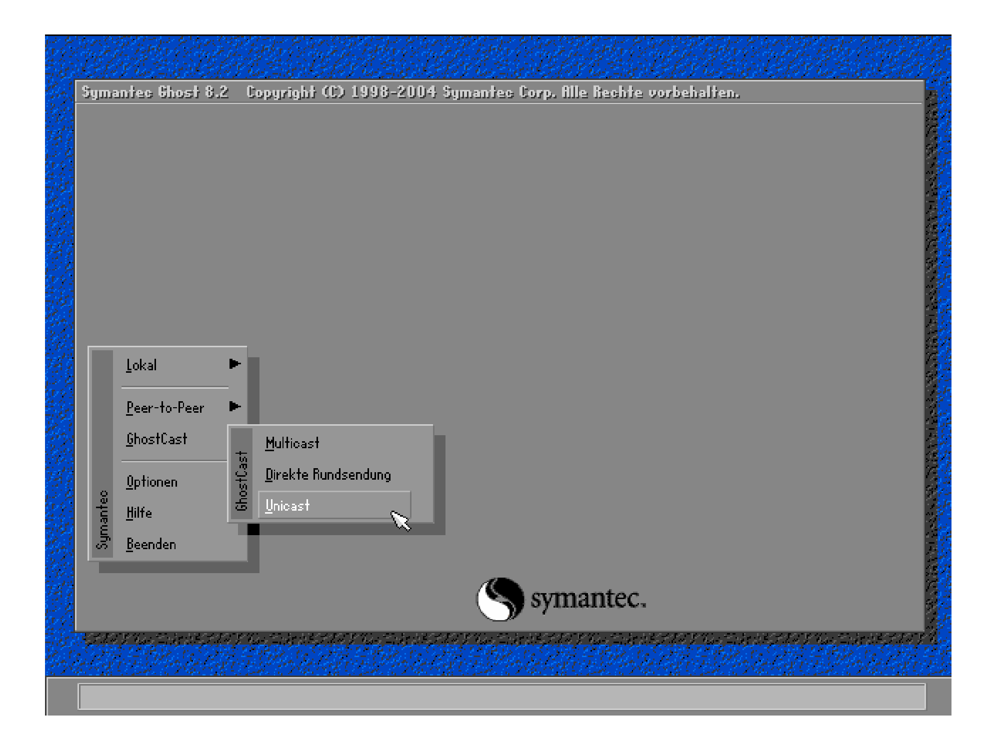
Innotek Virtual Box	http://www.virtualbox.org/
Microsoft Softgrid Application Virtualization	http://www.microsoft.com/systemcenter/softgrid/default.mspx
Microsoft Virtual PC 2007	http://www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx /
Microsoft Virtual Server 2005 R2	http://www.microsoft.com/germany/virtualserver/default.mspx
Parallels Workstation 2.2	http://www.parallels.com/products/workstation/
Sun Java Virtual Machine	http://www.java.com/de/download/manual.jsp
VMware Workstation (kostenplichtig)	http://www.vmware.com/de/products/ws/
Xen, Univerity of Cambridge	http://www.cl.cam.ac.uk/research/srg/netos/xen/
Xensource, Citrix XenServer ExpressEdition	http://www.citrixxenserver.com/products/Pages/XenExpress.aspx

12.2 Ghost Client am Beispiel 'restore' [Kap. 7.3.1]

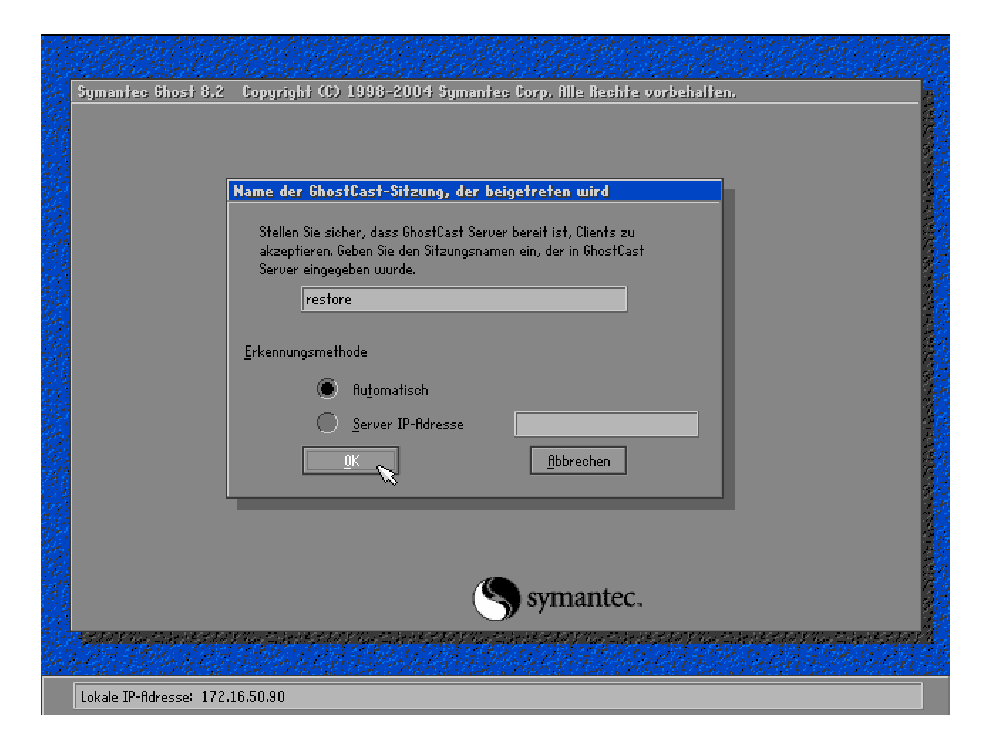
Start von Ghost (OK bestätigen)



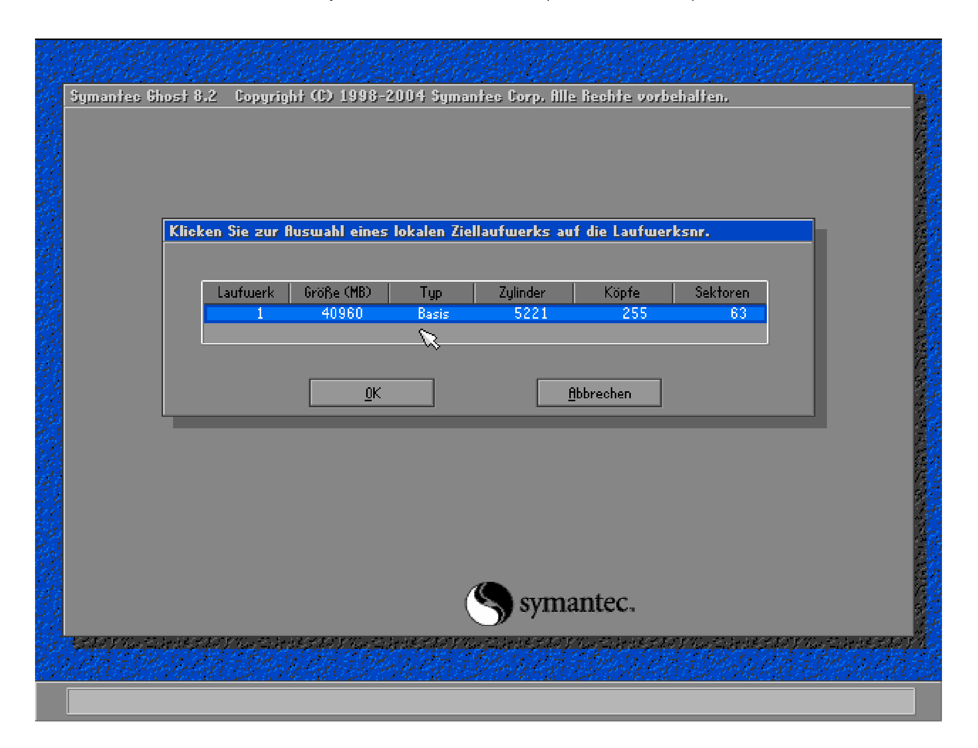
GhostCast-Modus auswählen (Unicast)



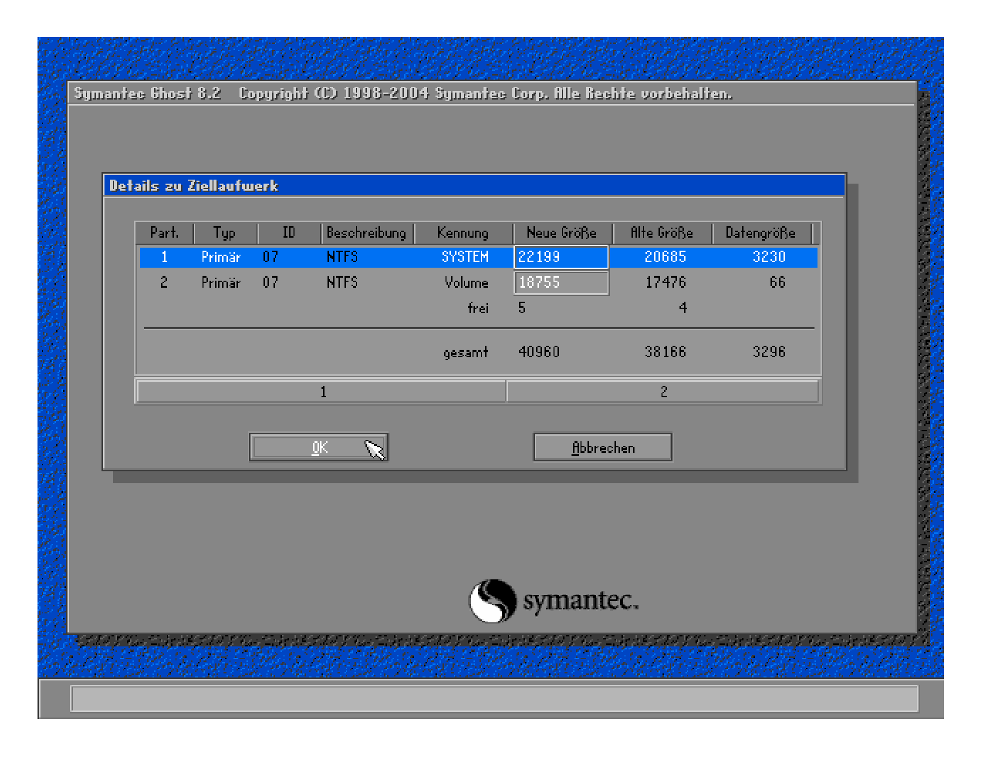
Sitzungsname eingeben (restore)



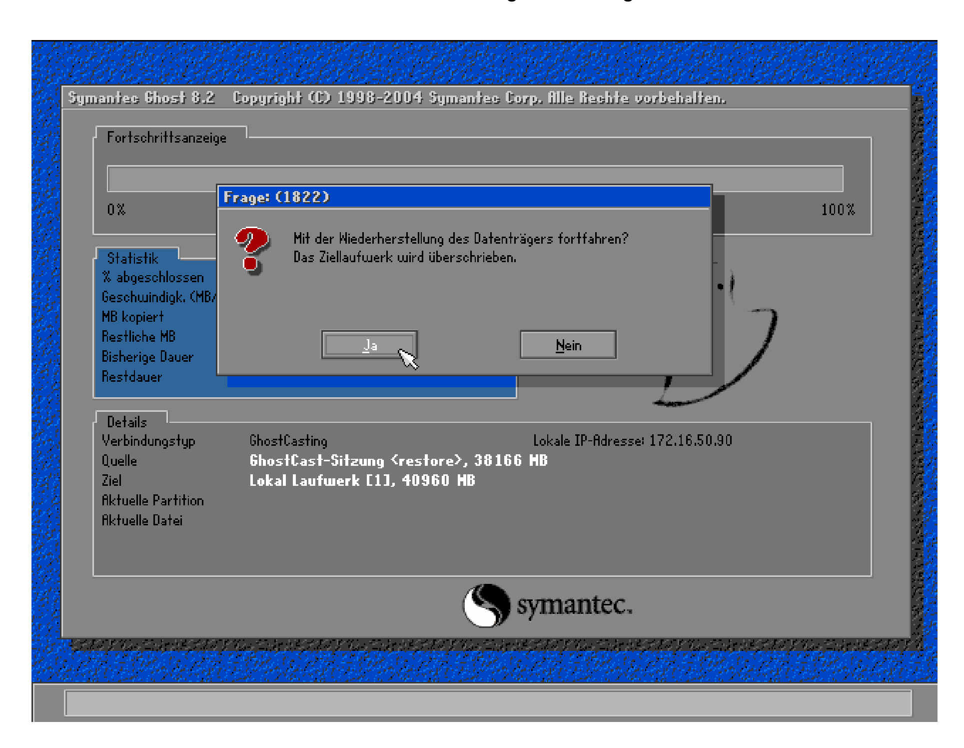
Festplatte auswählen (Laufwerk 1)



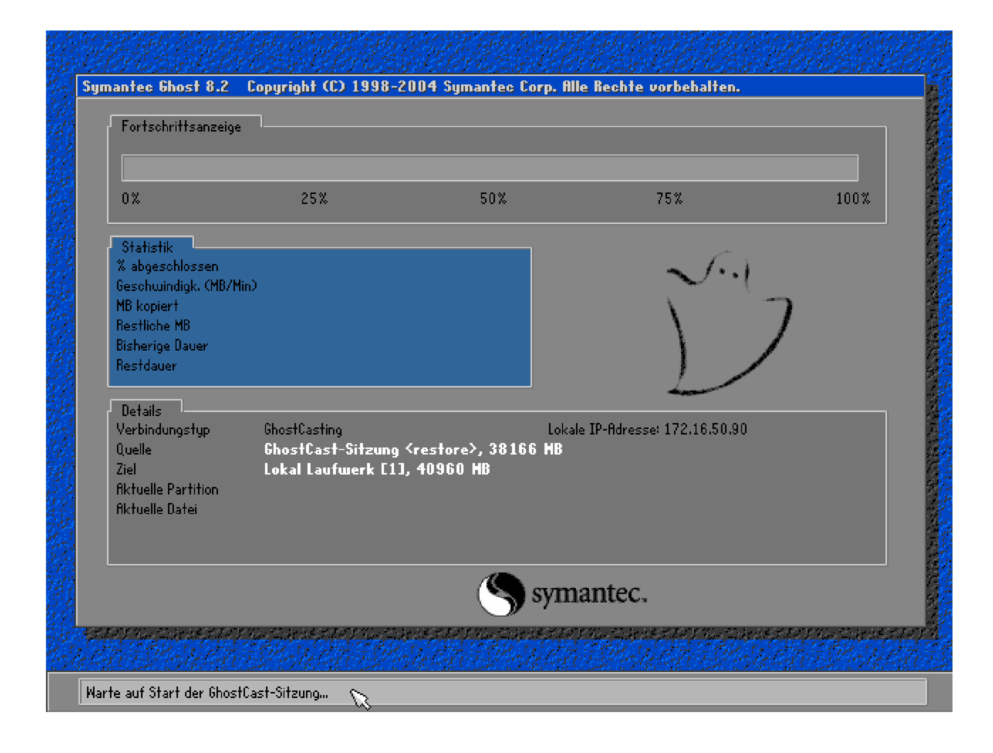
Partitionsgrößen anpassen (falls erforderlich)



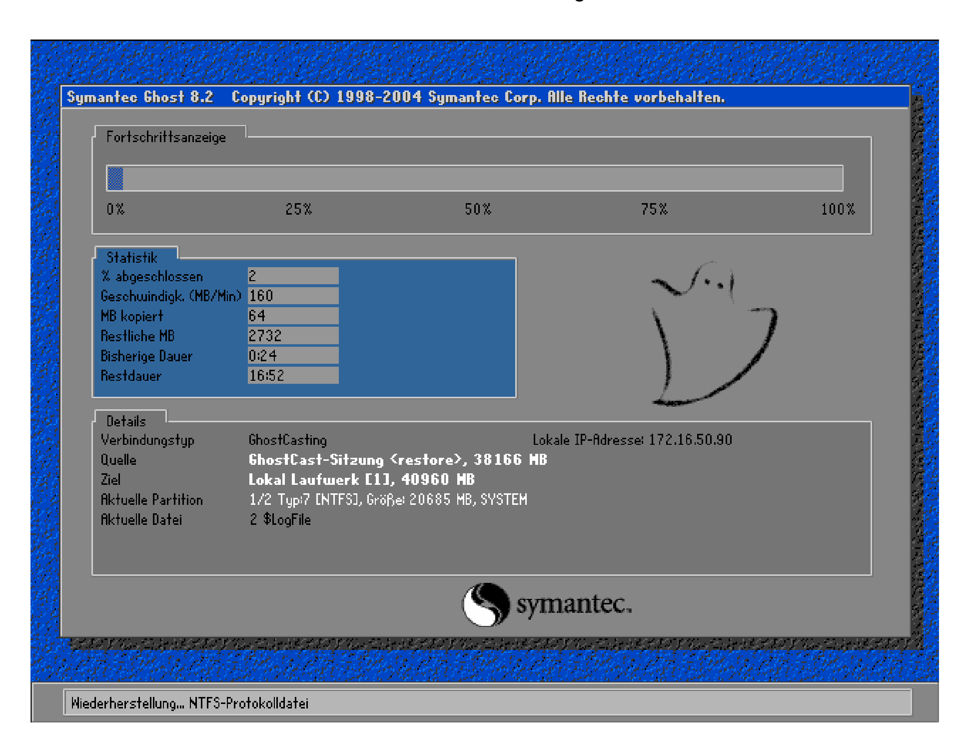
Sicherheitsabfrage bestätigen



Warten auf Start der Sitzung (GhostCast Server Sitzung "send" betätigen)

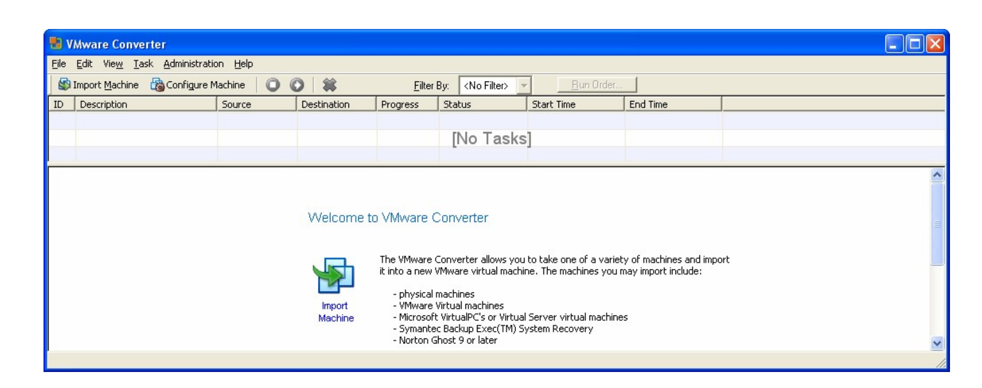


Fortschrittsanzeige



12.3 VMware Converter am Beispiel PC50090 [Kap. 8.4.1]

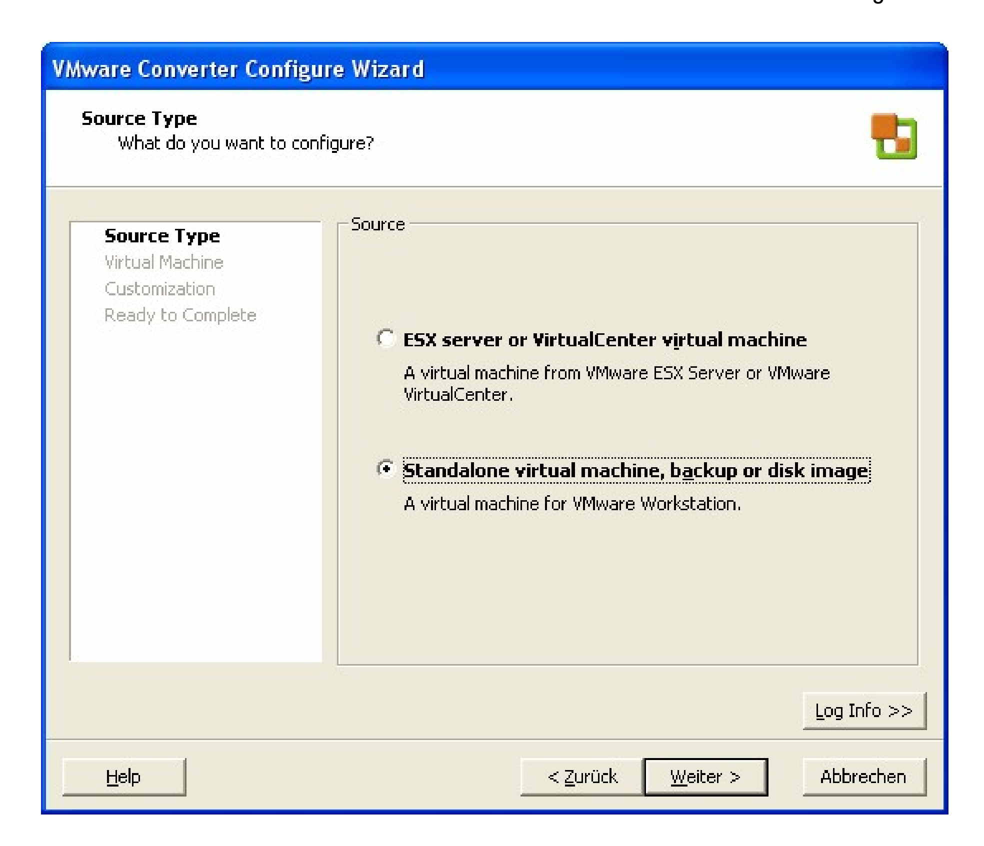
Startfenster des VMware Converter



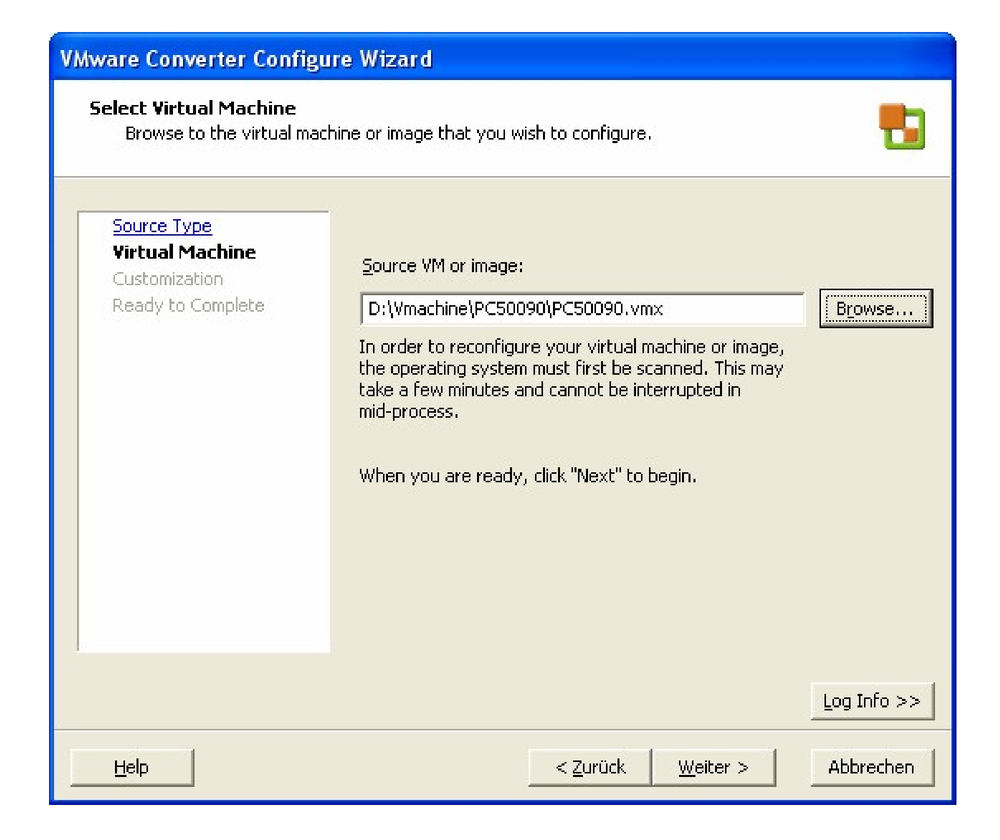
Eingangsfenster des Assistenten mit "Next" bestätigen



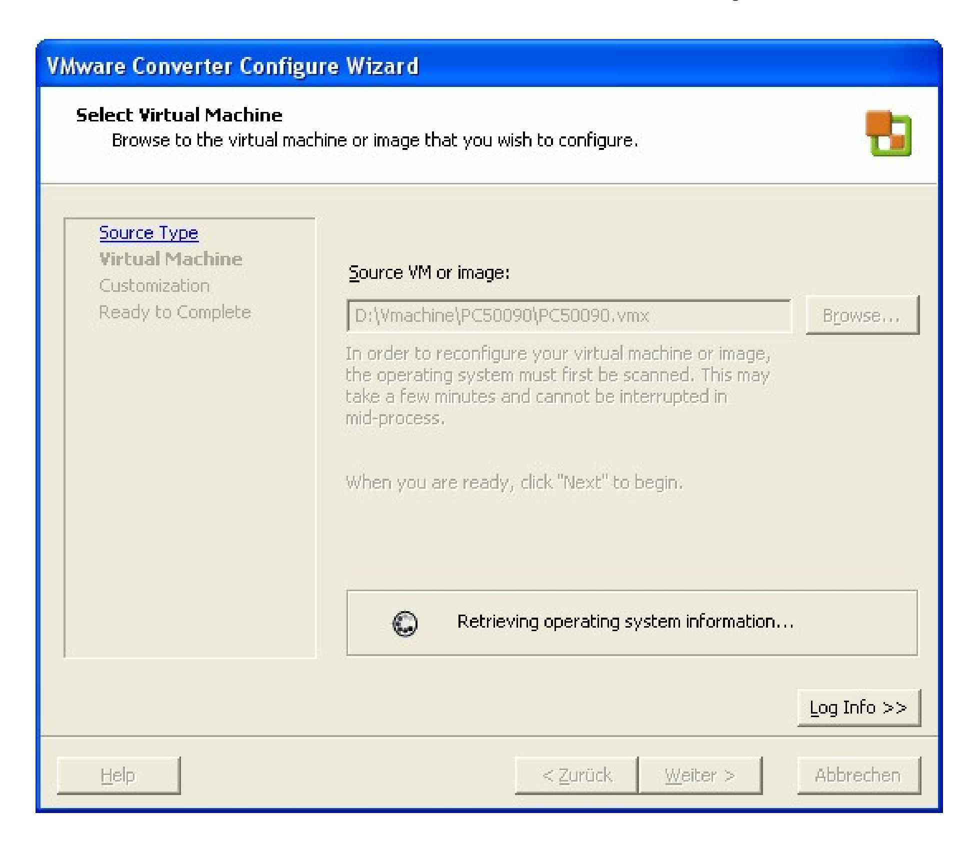
"Standalone virtual machine" auswählen und mit "Weiter" bestätigen



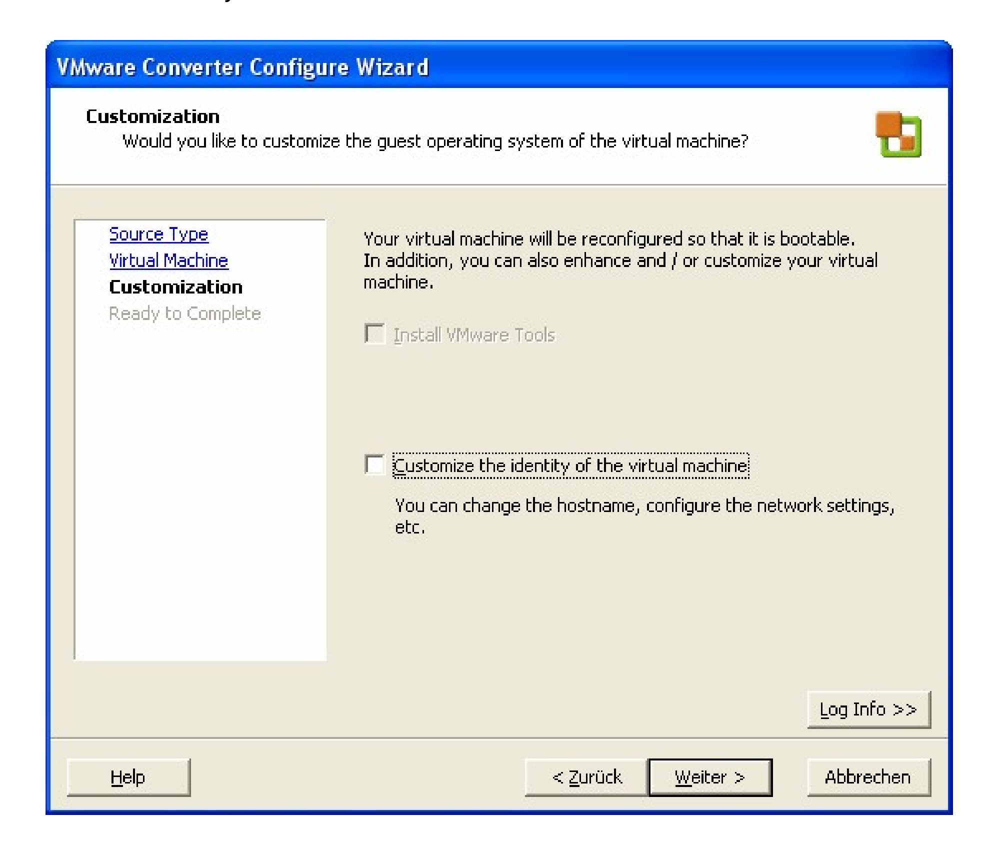
vmx-Datei des virtuellen PCs auswählen und mit "Weiter" bestätigen



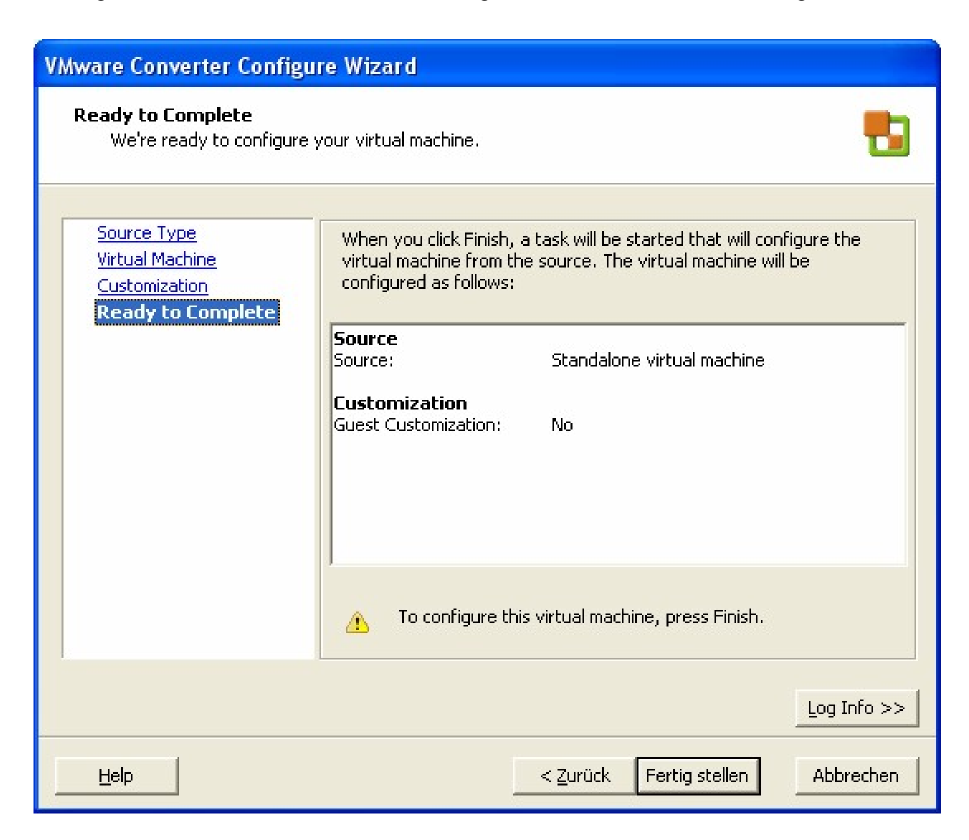
Statusfenster während der Konvertierung



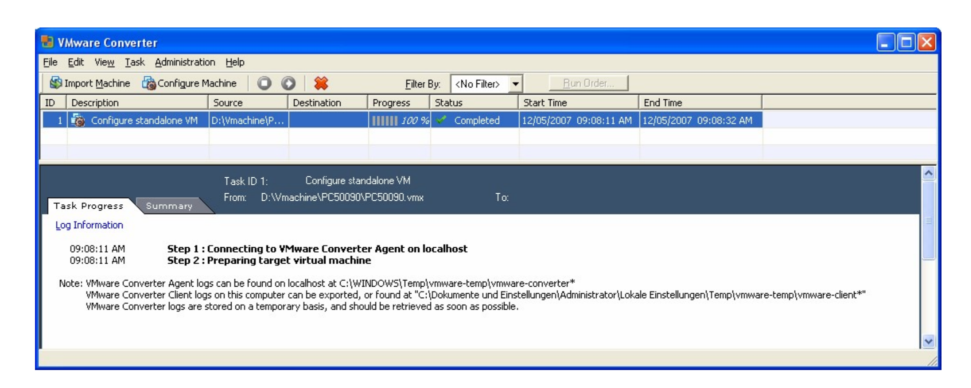
"Customize the identity of the virtual machine" nicht anhaken und mit "Weiter" bestätigen



Ergebnisfenster der Konvertierungsaktion, weiter mit "Fertig stellen"



Übersicht nach Abschluss der Konvertierung



12.4 Scripte

Einige Scripte wurden zwecks besserer Lesbarkeit umgebrochen. Die Reihenfolge der Scripts ist chronologisch, wie sie im vorherigen Text erwähnt worden sind. Soweit Scripte mehrfach, z.B. für verschiedene Standorte, vorhanden sind, werden an dieser Stelle nur exemplarisch einige Scripte ausführlich vorgestellt, soweit das für das Verständnis und die Arbeitsweise der Scripte notwendig ist.

12.4.1 Wake On LAN [Kapitel 7.3.3]

Die drei folgenden Scripte sind auf dem Server *SCOUTNG* unterhalb des Orderns *putty-scripte* abgelegt. In diesem Ordner liegt auch das benutzte Programm *plink.exe*. Mittels *plink.exe* wird eine SSH-Sitzung gestartet und der übergebene Befehl wird in einer remote-shell auf dem Zielhost ausgeführt.
Die beiden Scripte *pc59001* und *wakeall* liegen auf dem DHCP-Server in der Außenstelle. Der Inhalt der von *pc59001* aufgerufenen Datei *pc59001.mac* ist ebenfalls dargestellt.

PC59001_wol.cmd

```
@ECHO OFF
REM
..\GaAn_wol pc59001
```

wakeall.cmd

```
@ECHO OFF
REM
..\GaAn_wol wakeall
```

GaAn_wol.cmd

```
@ECHO OFF
REM
REM GaAn 172.19.0.0 mit DHCP-Server auf 172.19.1.1
SET redwall=172.19.1.1
IF %1%Test==Test GOTO NoPCName

REM
ECHO Folgender Befehl wird auf  %1% ausgefuehrt:
ECHO .
..\plink -ssh -pw password -v -l root  %redwall% sh /etc/wakeonlan/%1%
ECHO.
ECHO fertig
pause
GOTO ENDE

:NoPCName
ECHO.
ECHO Script nicht direkt aufrufen oder PC nicht gefunden.
ECHO.
PAUSE
GOTO ENDE

:ENDE
```

/etc/wakeonlan/pc59001

```
#! /bin/bash
WolHost=pc59001
ScriptPath=/etc/wakeonlan
MainPerlScript=wakeonlan

echo "*********************************************************************"
echo "Script $WolHost auf $HOSTNAME gestartet"
echo "Endgeraet $WolHost wird gesucht und per wake on lan gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$WolHost.mac
echo "Script $WolHost beendet."
echo "*********************************************************************"
```

/etc/wakeonlan/pc59001.mac

```
#pc59001
# MAC Broadcastadress
00:11:22:33:44:55 255.255.255.255
```

/etc/wakeonlan/wakeall

```
#! /bin/bash
WolHost=wakeall
ScriptPath=/etc/wakeonlan
MainPerlScript=wakeonlan

echo "##--##--##--##--##--##--##--##--##--##--##--##--##--##--##"
echo "Script $WolHost auf $HOSTNAME gestartet"
echo "Alle Endgeräte werden per wake on lan gestartet..."

Endgeraet=pc59001
echo "Endgeraet $Endgeraet wird gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$Endgeraet.mac

Endgeraet=pc59002
echo "Endgeraet $Endgeraet wird gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$Endgeraet.mac

Endgeraet=pc59003
echo "Endgeraet $Endgeraet wird gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$Endgeraet.mac

Endgeraet=pc59004
echo "Endgeraet $Endgeraet wird gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$Endgeraet.mac

Endgeraet=pc59005
echo "Endgeraet $Endgeraet wird gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$Endgeraet.mac

Endgeraet=pc59006
echo "Endgeraet $Endgeraet wird gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$Endgeraet.mac

Endgeraet=pc59007
echo "Endgeraet $Endgeraet wird gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$Endgeraet.mac

Endgeraet=pc59008
echo "Endgeraet $Endgeraet wird gestartet..."
perl $ScriptPath/$MainPerlScript -f $ScriptPath/$Endgeraet.mac

echo "Script $WolHost beendet."
echo "##--##--##--##--##--##--##--##--##--##--##--##--##--##--##"
```

12.4.2 Zusätzliche Automatisierung [Kapitel 7.4.1]

1login.cmd (Endgerät)

```
@start c:\install\Sysprep\W2kLockDesktop.exe
@ECHO OFF
REM HT/25.01.2008
REM 1login.cmd
REM
SET InstDir=c:\install
ECHO 1login.cmd
ECHO.

:MakeLokalAdmin
REM PCinst in lokale Administratoren aufnehmen
ECHO Gruppe der lokalen Administratoren anpassen...
@start /wait net localgroup administratoren KV\PCinst /add
@%InstDir%\Sysprep\wait 1

:AutoLoginOff
REM Autologin wieder deaktivieren
@start /wait reg import %InstDir%\Sysprep\autologon\autologin_ohne_username.reg
>nul
ECHO.
ECHO ------* Autologin deaktiviert
ECHO.
@%InstDir%\Sysprep\wait 1

:Inst
ECHO.
ECHO -------* Beliebige Anwendung installieren, Dauer: ca. 5-10 Minuten
ECHO.
REM @%InstDir%\Citrix\mps_setup.cmd
@start /wait %InstDir%\Sysprep\instservice.cmd
@start /wait %InstDir%\PCinst\pcinst.cmd
@start /wait %InstDir%\Sysprep\dsmodpc.cmd
@start /wait %InstDir%\session\regtlb \\DBSRV8\Session$\SmcServer.tlb
@%InstDir%\Sysprep\wait 1

:ENDE
ECHO Rechner wird neu gestartet...
:Shutdown
@start %InstDir%\Sysprep\shutdown.exe /L /R /T:10 "Reboot laeuft" /C
@%InstDir%\Sysprep\wait 1
:Clean
@start %InstDir%\Sysprep\2.cmd
EXIT
```

1login.cmd (Server)

```
REM @start c:\install\Sysprep\W2kLockDesktop.exe
@ECHO OFF
REM HT/09.02.2008
REM 1login.cmd
REM
SET InstDir=c:\install
ECHO 1login.cmd
ECHO.

:AutoLoginOff
REM Autologin wieder deaktivieren
@start /wait reg import %InstDir%\Sysprep\autologon\autologin_ohne_username.reg
>nul
ECHO.
ECHO ------* Autologin deaktiviert
ECHO.
@%InstDir%\Sysprep\wait 1

:Inst
ECHO.
ECHO -------* Beliebige Anwendung installieren, Dauer: ca. 5-10 Minuten
ECHO.
@start /wait %InstDir%\Sysprep\dsmodts.cmd
@start /wait %InstDir%\session\regtlb \\DBSRV8\Session$\SmcServer.tlb
@%InstDir%\Citrix\mps_setup.cmd
@%InstDir%\Sysprep\wait 1

:ENDE
ECHO Rechner wird neu gestartet...
:Shutdown
@start %InstDir%\Sysprep\shutdown.exe /L /R /T:10 "Reboot laeuft" /C
@%InstDir%\Sysprep\wait 1
:Clean
@start %InstDir%\Sysprep\2.cmd
EXIT
```

instservice.cmd

```
@ECHO OFF
@instsrv pcinst C:\Windows\System32\srvany.exe -a KV\PCinst -p pcinst
@REG ADD HKLM\SYSTEM\CurrentControlSet\Services\PCinst /v Description /t REG_SZ
/d "Installerdienst der KVMYK"
@REG ADD HKLM\SYSTEM\CurrentControlSet\Services\PCinst\Parameters
@REG ADD HKLM\SYSTEM\CurrentControlSet\Services\PCinst\Parameters /v Application
/t REG_SZ /d "C:\Install\PCinst\install.cmd"

:ENDE
EXIT
```

pcinst.cmd

```
@REM #!%ComSpec%
@ECHO OFF
REM ## autor: ht
REM ## date: 20071130
REM ## name: pcinst.cmd
REM ## location: \install\pcinst
REM ## zweck: bei Bedarf lokale Anwendung nach erster Anmeldung installieren
REM ## zweck: arbeitet als lokaler administrator
REM ## function: install local application if needed
REM ## function: works as local administrator
REM ## version: 1

REM ## set environment
SET scriptname=pcinst.cmd
SET ScriptPath=%SystemDrive%\install\PCinst
SET scripttoexecute=%SystemDrive%\install\PCinst\install.cmd
SET logfile=%ScriptPath%\%scriptname%.log
REM ## end environment

REM ## starting logfile
ECHO Script %scriptname% started at >%logfile%
date /T  >>%logfile%
time /T >>%logfile%
ECHO ********************** >>%logfile%

REM ***************************************
ECHO -* searching for %scripttoexecute% to execute... >>%logfile%
ECHO -* searching for %scripttoexecute% to execute...
IF EXIST %scripttoexecute% ECHO -* %scripttoexecute% found, please wait while
executing >>%logfile%
IF EXIST %scripttoexecute% ECHO -* %scripttoexecute% found, please wait while
executing
IF EXIST %scripttoexecute% start /wait %scripttoexecute%
IF NOT EXIST %scripttoexecute% ECHO %scripttoexecute% not found >>%logfile%
IF NOT EXIST %scripttoexecute% ECHO %scripttoexecute% not found
GOTO END

REM ***************************************
:END
ECHO *** END %scriptname%. *** >>%logfile%
ECHO.>>%logfile%
ECHO *** END %scriptname%. ***
ECHO.

:ENDE
EXIT
```

dsmodpc.cmd

```
@ECHO OFF
SET DsModDir=c:\install\sysprep
SET
grp=CN=bEndgeraeteMaschinen,OU=Endgeraete,OU=Systemgruppen,OU=Gruppen,OU=KVMYK,D
C=kv,DC=myk,DC=intern
SET
mbr=CN=%ComputerName%,OU=Standard,OU=Computer,OU=KVMYK,DC=kv,DC=myk,DC=intern
SET usr=administrator
SET pwd=password
SET srv=DCKV01
SET dom=KV

%DsModDIr%\dsget group "%grp%" -members -u %usr% -p %pwd% -d %dom% -c -q |find
"%ComputerName%" 2>&1 >nul
IF %ErrorLevel% == 0 GOTO IsMember
IF %ErrorLevel% == 1 GOTO NotMember
ECHO Unknown Error occurred!!
GOTO END

:IsMember
ECHO.
ECHO %ComputerName% is Member of Group %grp%
ECHO Nothing to do :-)
GOTO END

:NotMember
ECHO.
ECHO %ComputerName% is not member of Group %grp%
ECHO changing...
%DsModDir%\dsmod group "%grp%" -addmbr "%mbr%" -u %usr% -p %pwd% -d %dom% -c -q
IF %ErrorLevel% == 0 GOTO IsMemberOk
ECHO Error occurred!!
GOTO END

:IsMemberOk
ECHO modify group "%grp%" with "%ComputerName%" successful
ECHO.
GOTO END

:END
gpupdate /force
EXIT
```

dsmodts.cmd

```
@ECHO OFF
SET DsModDir=c:\install\sysprep
SET
grp=CN=bCitrixServer,OU=Server,OU=Systemgruppen,OU=Gruppen,OU=KVMYK,DC=kv,DC=myk
,DC=intern
SET
mbr=CN=%ComputerName%,OU=Standard,OU=Computer,OU=KVMYK,DC=kv,DC=myk,DC=intern
SET usr=administrator
SET pwd=password
SET srv=DCKV01
SET dom=KV

%DsModDIr%\dsget group "%grp%" -members -u %usr% -p %pwd% -d %dom% -c -q |find
"%ComputerName%" 2>&1 >nul
IF %ErrorLevel% == 0 GOTO IsMember
IF %ErrorLevel% == 1 GOTO NotMember
ECHO Unknown Error occurred!!
GOTO END

:IsMember
ECHO.
ECHO %ComputerName% is Member of Group %grp%
ECHO Nothing to do :-)
GOTO END

:NotMember
ECHO.
ECHO %ComputerName% is not member of Group %grp%
ECHO changing...
%DsModDir%\dsmod group "%grp%" -addmbr "%mbr%" -u %usr% -p %pwd% -d %dom% -c -q
IF %ErrorLevel% == 0 GOTO IsMemberOk
ECHO Error occurred!!
GOTO END

:IsMemberOk
ECHO modify group "%grp%" with "%ComputerName%" successful
ECHO.
GOTO END

:END
gpupdate /force
EXIT
```

2.cmd

```
@ECHO OFF
SET InstDir=c:\install
@del %InstDir%\Sysprep\1login.cmd
@del %InstDir%\Sysprep\instservice.cmd
@del %InstDir%\Sysprep\dsmod*.cmd
@rd /s /q %InstDir%\Sysprep\Autologon
exit
```

autologin.cmd

```
@ECHO OFF

:SetRunOnce
REM RunOnceKey setzen
ECHO RunOnce-Key setzen
reg import C:\Sysprep\autologon\RunOnce_ip2name.reg
ECHO.

:SetDefaultPwd
REM Administratorkennwort aendern
ECHO Administratorkennwort aendern
CALL C:\sysprep\pspasswd administrator password
ECHO.

:AutoLoginOn
REM Autologin aktivieren
ECHO Autologin aktivieren
ECHO.
C:\Sysprep\Autologon\autologon administrator pc60xyz password

:ReleaseIp
Ipconfig /release
pause
```

ip2name.cmd

```
START c:\install\sysprep\W2KLockDesktop.exe
@REM #!%ComSpec%
@ECHO OFF
REM ## autor: ht
REM ## date: 20071212
REM ## name: ip2name.cmd
REM ## location: \sysprep
REM ## Zweck: auf ip warten, dann perl-script ip2name.pl aufrufen
REM ## function: wait for ip-adress, then start perl-script ip2name.pl
REM ## needs: 'wait.exe', perl and ip2name.pl
REM ## version: 1
REM ## only for german OS!!!

REM ## set environment
SET ScriptName=ip2name.cmd
SET Logfile=%SystemDrive%\temp\%scriptname%.log

SET SysprepDir=c:\sysprep
REM ## end environment

CLS
ECHO.
ECHO Bitte nicht mit der Maus arbeiten, der Rechner wird automatisch konfiguriert.
ECHO Nach Abschluss bootet der Rechner von alleine.
ECHO Hardwarekonfiguration (Plug'N'Play) in Arbeit...
ECHO.
ECHO Auf IP-Adresse warten...
ipconfig /renew >nul
:WaitForIp
%SysprepDir%\wait 10 >nul
ipconfig >%SysprepDir%\connect
type %SysprepDir%\connect | find "kein Vorgang" 2>&1 >nul
IF %ErrorLevel% == 0 ECHO -* Netzwerkkabel nicht gesteckt...
IF %ErrorLevel% == 0 GOTO WaitForIp

type %SysprepDir%\connect | find "IP-Konfiguration" 2>&1 >nul
IF %ErrorLevel% == 0 GOTO CheckIp
```

```
GOTO WaitForIp

:CheckIp
type %SysprepDir%\connect | find "IP-Adresse" 2>&1 >nul
IF %ErrorLevel% == 1 ECHO -* noch keine IP-Adresse erhalten
IF %ErrorLevel% == 1 GOTO WaitForIp
IF %ErrorLevel% == 0 GOTO IPok
IF EXIST %SysprepDir%\Testmode.yes ECHO Errorlevel = %ErrorLevel%
GOTO WaitForIp

:IPok
ECHO ok
ECHO.
ECHO Der Rechner macht nach 3 Sekunden weiter...bitte Geduld.
%SysprepDir%\wait 3
IPCONFIG | find "IP-Adresse" 2>&1

:SrchSysprep2
ECHO.
ECHO Jetzt sysprep2.inf auswerten
IF EXIST %SysprepDir%\sysprep2.inf GOTO Input
ECHO --* keine Datei %SysprepDir%\sysprep2.inf gefunden
GOTO ENDE

:Input
IF EXIST %SysprepDir%\input.txt del %SysprepDir%\input.txt >nul
ECHO ---* vorhandene input.txt wurde geloescht...
IPCONFIG /all>%SysprepDir%\input.txt

IF EXIST %SysprepDir%\INPUT.TXT GOTO INPUTOK
ECHO.
ECHO ---* Datei %SysprepDir%\INPUT.TXT nicht gefunden
ECHO.
pause
GOTO ENDE

:INPUTOK
IF EXIST %SysprepDir%\Sysprep.inf del %SysprepDir%\sysprep.inf >nul

:StartScript
%SysprepDir%\perl\bin\perl %SysprepDir%\ip2name.pl
ECHO.

:CheckTestMode
IF EXIST %SysprepDir%\Testmode.yes GOTO Testbetrieb >nul

:ResetPwd
CALL C:\sysprep\pspasswd administrator >nul
ECHO.
ECHO --* Lokales Administratorkennwort geaendert
%SysprepDir%\wait 2

:MakeSysprep
ECHO.
ECHO Der Rechner wird nun fuer den naechsten Neustart vorbereitet...
%SysprepDir%\wait 2
%SysprepDir%\sysprep -mini -activated -quiet -reseal -reboot
GOTO ENDE

:Testbetrieb
ECHO.
ECHO -------*****************************************--------
ECHO -------* Testbetrieb Sysprep nicht ausgefuehrt. --------
ECHO -------*****************************************--------
ECHO.

:ENDE
```

mps_setup.cmd

```
C:\Install\Citrix\unattendedinstall.exe "C:\Install\Citrix\mps.msi"
"C:\Install\Citrix\mps_setup.txt" CTX_MF_ADD_ANON_USERS=NO
 CTX_MF_CREATE_REMOTE_DESKTOP_USERS=DoNothing
CTX_ADDLOCAL=MetaFrame_XP,CTX_MF_MetaFrame_Core,CTX_MF_IMA_Core,
CTX_MF_ICA_Shell_Editor,CTX_SMA,CTX_MF_CTXCPU,CTX_MF_CTXSFO,PN_AGENT,PN_ENGINE,
 CTX_MF_IM_Service,CTX_MF_RM,WMI CTX_MF_ONLY_LAUNCH_PUBLISHED_APPS=Yes
 SERVER_LOCATION=http://pnagent.kv.myk.intern ENABLE_SSON=YES
```

12.4.3 vm-do-backup.cmd [Kapitel 8.3]

```
@REM #!%ComSpec%
@ECHO OFF
REM ## autor: ht
REM ## date: 20080213
REM ## name: vm-do-backup.cmd
REM ## location: \batch
REM ## Zweck: backupscript fuer vmthost01 und vmthost02
REM ## function: control script vmthost01 and vmthost02
REM ## needs: xcopy
REM ## version: 1

REM ## set environment
SET scriptpath=c:\batch
SET scriptname=vm-do-backup.cmd
SET logfile=%SystemDrive%\Batch\%scriptname%.log
SET logfile=c:\batch\%scriptname%-%1%.log
SET vmMachinePath=z:\vmachine
SET vmbackuptarget=d:\vmachinebackup
SET dayofweek=%1
SET RemoteHost=172.20.7.101
IF NOT %2.==. SET RemoteHost=%2%
IF NOT %2.==. ECHO remoteHost=%RemoteHost%

SET BenutzerName=Administrator
SET pwd=administrator
SET xcopyoptions=/c /s /e /r /h /y /f
REM ## end environment

IF %1empty==empty GOTO ShowSyntax

REM ## starting logfile
ECHO Script %scriptname% started at >%logfile%
date /T  >>%logfile%
time /T >>%logfile%
ECHO ********************** >>%logfile%
IF NOT %1empty==empty ECHO Parameter 1 = %1 (Dayofweek) >>%logfile%
ECHO ********************** >>%logfile%

IF NOT EXIST Z: GOTO NOZ
NET USE Z: /DELETE >>%logfile%
:NOZ
IF NOT EXIST Y: GOTO NOY
NET USE Y: /DELETE >>%logfile%
:NOY

ECHO try to connect z: to \\%RemoteHost%\vmdaten >>%logfile%
ECHO try to connect z: to \\%RemoteHost%\vmdaten
NET USE z: \\%RemoteHost%\vmdaten /USER:%BenutzerName% %pwd% >>%logfile%
IF NOT EXIST Z:\vmachine GOTO ErrorZ

ECHO try to connect y: to \\%RemoteHost%\vmbackup >>%logfile%
ECHO try to connect y: to \\%RemoteHost%\vmbackup
NET USE y: \\%RemoteHost%\vmbackup /USER:%BenutzerName% %pwd% >>%logfile%
IF NOT EXIST Y:\vmachinebackup GOTO ErrorY
```

```
REM ***************************************
ECHO -* starting xcopy... >>%logfile%
ECHO -* starting xcopy...

REM first: RemoteHost->local
IF EXIST %vmbackuptarget%\%DayOfWeek% rd /s /q %vmbackuptarget%\%DayOfWeek%
IF NOT EXIST %vmbackuptarget%\%DayOfWeek% md %vmbackuptarget%\%DayOfWeek%
ECHO xcopy %vmMachinePath% %vmbackuptarget%\%DayOfWeek% %xcopyoptions%
>>%logfile%
ECHO xcopy %vmMachinePath% %vmbackuptarget%\%DayOfWeek% %xcopyoptions%
xcopy %vmMachinePath% %vmbackuptarget%\%DayOfWeek% %xcopyoptions% >>%logfile%
ECHO. >>%logfile%

REM second: local->RemoteHost
SET vmMachinePath=d:\vmachine
SET vmbackuptarget=y:\vmachinebackup
IF EXIST %vmbackuptarget%\%DayOfWeek% rd /s /q %vmbackuptarget%\%DayOfWeek%
IF NOT EXIST %vmbackuptarget%\%DayOfWeek% md %vmbackuptarget%\%DayOfWeek%
ECHO xcopy %vmMachinePath% %vmbackuptarget%\%DayOfWeek% %xcopyoptions%
>>%logfile%
ECHO xcopy %vmMachinePath% %vmbackuptarget%\%DayOfWeek% %xcopyoptions%
xcopy %vmMachinePath% %vmbackuptarget%\%DayOfWeek% %xcopyoptions% >>%logfile%

GOTO END

:ErrorZ
ECHO.
ECHO could not connect to z: >>%logfile%
ECHO could not connect to z:
GOTO END

:ErrorY
ECHO.
ECHO could not connect to y: >>%logfile%
ECHO could not connect to y:
GOTO END

:ShowSyntax
ECHO.
ECHO Missing Parameter
ECHO.
ECHO Syntax: %scriptname% DayOfWeek [RemoteHostName or IP-Adress]
ECHO Example 1: %scriptname% thuesday [VMTHOST02]
ECHO Example 2: %scriptname% thuesday [192.268.75.92]
GOTO ENDE

REM ***************************************
:END
ECHO.>>%logfile%
date /T  >>%logfile%
time /T >>%logfile%
ECHO *** END %scriptname% *** >>%logfile%
ECHO *** END %scriptname% ***
ECHO.
IF EXIST Y: NET USE Y: /D
IF EXIST Z: NET USE Z: /D
IF NOT %tst%==yes EXIT

:ENDE
EXIT /B
```

12.4.4 vmpc.cmd [Kap. 8.3.2]

```
@ECHO OFF
REM vmpc.cmd
REM HT/20080216
REM install virtuall machines
REM 1. searching installed vm (server or workstation) in %VM%
REM 2. copy all needed files from %ImgSrc%\%VMPCName% to %ImgTgt%\%VMPCName%
REM 3. call perl-script ip2name.pl with parameters '-vm' and 'MAC' to modify
vmpc.vmx
REM 4. start vmware.exe with modified vmx.file
REM
REM Prereq: -DHCP-Reservation for desired MAC-Adress
REM         -read-access to %ImgSrc$
REM         -write-access to %ImgTgt$
REM         -installed VMWare Server 1.x or Workstation 6.x
REM         -perl.script ip2name.pl and ini.file ip2name.ini
REM
SET Script=vmpc.cmd
REM # define image source and target
REM SET ImgSrc=\\Imagesrv\vmpc$ (default)
REM SET ImgTgt=c:\programme\vmware\vmachine (default)
SET ImgSrc=\\Imagesrv\\vmpc$
SET VMserver=%ProgramFiles%\vmware\vmware server
SET VMwkst=%ProgramFiles%\vmware workstation
SET ImgTgt=d:\vmachine
SET ActualDrive=%~d1

REM # define default pc-name, displayed by vmware
REM SET VMPCName=vmpc (default)
SET VMPCName=vmpc

REM # define source config-file to copy
REM SET VMPCvmx=vmpc.vmx (default for pc)
REM SET VMPCvmx=vmts.vmx (default for ts)
SET VMPCvmx=%VMPCName%.vmx

REM #define path to perl binaries and default script
SET PerlPath=%ImgSrc%\perl\bin
SET DefaultScript=%ImgSrc%\ip2name.pl
SET Parameter=-vm

REM #check needed first parameter MAC-Adress
REM #format dd:dd:dd:dd:dd:dd
if %1.%==. GOTO MissingParameter
SET MAC=%1%

REM #making network connection
IF EXIST B: NET USE B: /DELETE >NUL
NET USE B: %ImgSrc%
if ERRORLEVEL == 0 GOTO NetConnectOK
ECHO Error while connecting network drive B: to %ImgSrc%
ECHO ending script
GOTO ENDE

:NetConnectOK
REM #copy files
ECHO searching for VMware Server...[%VMserver%]
if exist "%VMserver%\vmware.exe" GOTO VMwareServerFound
ECHO ...not found
ECHO.
ECHO searching for VMware Workstation...[%VMwkst%]
if exist "%VMwkst%\vmware.exe" GOTO VMwareWkstFound
ECHO Vmware not found...ending
GOTO InstVM
```

```
:VMwareServerFound
ECHO ...found
SET VM=%VMserver%
DIR "%VM%\vmware.exe"
ECHO.
GOTO CopyVM

:VMwareWkstFound
ECHO ...found
SET VM=%VMwkst%
DIR "%VM%\vmware.exe"
ECHO.
GOTO CopyVM

:CopyVM
ECHO starting copy files...
if not exist "%ImgTgt%\%VMPCName%" md "%ImgTgt%\%VMPCName%"
if exist "%ImgTgt%\%VMPCName%" copy %ImgSrc%\%VMPCName%\*.*
"%ImgTgt%\%VMPCName%" /y >nul
if %ErrorLevel% == 0 GOTO CopyOk
ECHO Error while copy %ImgSrc%\%VMPCName%\*.* to "%ImgTgt%\%VMPCName%"
GOTO ENDE
:CopyOk
ECHO copy %ImgSrc%\*.* to "%ImgTgt%\%VMPCName%": successful

REM #modify files
if not exist %PerlPath%\perl.exe GOTO MissingPerl
ECHO.
ECHO ****************************************************************
ECHO starting %DefaultScript% with parameter
ECHO %Parameter% %MAC% "%ImgTgt%\%VMPCname%\%VMPCvmx%"
ECHO ****************************************************************
ECHO.
B:
REM %PerlPath%\perl.exe %DefaultScript% %Parameter% %MAC%
"%ImgTgt%\%VMPCname%\%VMPCvmx%"
%PerlPath%\perl.exe %DefaultScript% %Parameter% %MAC%
"%ImgTgt%\%VMPCname%\%VMPCvmx%"
if %ErrorLevel% == 0 GOTO ScriptOk
ECHO Error while executing %DefaultScript% %Parameter% %MAC%
"%ImgTgt%\%VMPCname%\%VMPCvmx%"
GOTO ENDE

:ScriptOk
ECHO.
ECHO now starting VMware with the new machine...
start "VMware" /D"%VM%" vmware.exe -x -c "%ImgTgt%\%VMPCname%\%VMPCvmx%"
GOTO ENDE

:MissingPerl
ECHO.
ECHO perl not found at %PerlPath%
GOTO ENDE

:MissingParameter
ECHO.
ECHO missing parameter MAC-Adress
ECHO use dd:dd:ddd:d:dd:dd
ECHO example: %Script% 00:0c:29:12:34:56
ECHO.
GOTO ENDE

:InstVM
ECHO.
ECHO If you want to install 'VMware Server' call %ImgSrc%\silentinstvm.cmd
ECHO.
GOTO ENDE
```

```
:ENDE
%ActualDrive%
IF EXIST B: NET USE B: /DELETE >NUL
exit /b
```

12.5 Beispieldateien *sysprep.inf*

12.5.1 *sysprep.inf* (Server)

```
[Unattended]
    OemSkipEula=Yes
    InstallFilesPath=C:\sysprep\i386
    TargetPath=\WINDOWS
    OEMPnPDriversPath = "PnPDrvrs\PC-Type\Graphic\;PnPDrvrs\VMware\Graphic\"
    OemPreinstall = Yes
    UpdateServerProfileDirectory=0

[GuiUnattended]
    AdminPassword=password
    EncryptedAdminPassword=NO
    OEMSkipRegional=1
    OEMDuplicatorstring=PC60xyz-Baseimage-Sysprep-11122007
    TimeZone=110
    OemSkipWelcome=1
    AutoLogon=Yes
    AutoLogonCount=1

[BenutzerData]
    ProductKey=AAAAA-12345-BBBBB-67890-CCCCC
    FullName="FullName"
    OrgName="Organization"
    ComputerName=

[Identification]
    JoinDomain=DomainName
    DomainAdmin=AdminBenutzer
    DomainAdminPassword=password
    MachineObjectOU="OU = Standard,OU = Computer, DC = DomainName, DC = org"

[Networking]

[GuiRunOnce]
    Command0=C:\Install\Sysprep\1login.cmd

[sysprepcleanup]
Service=pci
Service=isapnp
Upperfilter=AGP440
Service=ftdisk
Service=swenum
Service=update
Service=mssmbios
Service=dmio
Service=rdpdr
Upperfilter=mouclass
Service=TermDD
Upperfilter=kbdclass
Upperfilter=ALIM1541
Upperfilter=AMDAGP
Upperfilter=AGPCPQ
Upperfilter=SISAGP
Upperfilter=VIAAGP
Service=aha154x
Service=sparrow
Service=adpu160m
Service=aic78u2
Service=aic78xx
Service=dpti2o
Service=cpqarray
```

```
Service=sym_hi
Service=mraid35x
Service=perc2
Lowerfilter=perc2hib
Service=hpn
Service=symc8xx
Service=sym_u3
Service=symc810
Service=ql10wnt
Service=ql1080
Service=ql1240
Service=ql1280
Service=ql12160
Service=i2omp
Service=cd20xrnt
Service=cbidf
Service=dac2w2k
Service=dac960nt
Service=asc3550
Service=asc
Service=asc3350P
Service=ABP480N5
Service=amsint
Service=ini910u
Service=ultra
Service=pciide
Service=intelide
Service=atapi
Service=aliide
Service=cmdide
Service=viaide
Service=toside
```

12.5.2 *sysprep.inf* (Endgerät)

```
[Unattended]
    OemSkipEula=Yes
    InstallFilesPath=C:\sysprep\i386
    TargetPath=\WINDOWS
    OEMPnPDriversPath = "PnPDrvrs\PC-Type\Graphic\;PnPDrvrs\VMware\Graphic\"
    OemPreinstall = Yes

[GuiUnattended]
    AdminPassword=password
    EncryptedAdminPassword=NO
    OEMSkipRegional=1
    OEMDuplicatorstring=PC60xyz-Baseimage-Sysprep-11122007
    TimeZone=110
    OemSkipWelcome=1
    AutoLogon=Yes
    AutoLogonCount=1

[BenutzerData]
    ProductKey=AAAAA-12345-BBBBB-67890-CCCCC
    FullName="FullName"
    OrgName="Organization"
    ComputerName=

[Identification]
    JoinDomain=DomainName
    DomainAdmin=AdminBenutzer
    DomainAdminPassword=password
    MachineObjectOU="OU = Standard,OU = Computer, DC = DomainName, DC = org"

[Networking]

[GuiRunOnce]
```

```
    Command0=C:\Install\Sysprep\1login.cmd

[sysprepcleanup]
Service=pci
Service=isapnp
Upperfilter=AGP440
Service=ftdisk
Service=swenum
Service=update
Service=mssmbios
Service=dmio
Service=rdpdr
Upperfilter=mouclass
Service=TermDD
Upperfilter=kbdclass
Upperfilter=ALIM1541
Upperfilter=AMDAGP
Upperfilter=AGPCPQ
Upperfilter=SISAGP
Upperfilter=VIAAGP
Service=aha154x
Service=sparrow
Service=adpu160m
Service=aic78u2
Service=aic78xx
Service=dpti2o
Service=cpqarray
Service=sym_hi
Service=mraid35x
Service=perc2
Lowerfilter=perc2hib
Service=hpn
Service=symc8xx
Service=sym_u3
Service=symc810
Service=ql10wnt
Service=ql1080
Service=ql1240
Service=ql1280
Service=ql12160
Service=i2omp
Service=cd20xrnt
Service=cbidf
Service=dac2w2k
Service=dac960nt
Service=asc3550
Service=asc
Service=asc3350P
Service=ABP480N5
Service=amsint
Service=ini910u
Service=ultra
Service=pciide
Service=intelide
Service=atapi
Service=aliide
Service=cmdide
Service=viaide
Service=toside
```

12.6 Beispieldateien ip2name.ini

Bei den beiden folgenden INI-Dateien sind sämtliche Kommentarzeilen entfernt worden.

12.6.1 *ip2name.ini (Server)*

```
[MODE]
GetNameMode=ip
FallbackMode=1

[SOURCE]
SysprepTpl=c:\sysprep\sysprep2.inf
SysprepSrchStr=ComputerName=

[TARGET]
Sysprep=c:\sysprep\sysprep.inf
Net=C
HostMask=11
HostMaskCheck=0
Suffix=CTXSRV
CmpNameLength=15

[LOCAL]
LocalInputFile=c:\sysprep\ip2name.txt

[HTTP]
HTTPHost=www.ip2name.org/
HTTPurl=ip2name.txt

[FTP]
FTPHost=ftp.ip2name.org/
FTPurl=ip2name.txt

[MAC]
MacInputFile=c:\sysprep\input.txt
MacFileSearchString=Physikalische Adresse . . . . . . :
MACSize=17

[VM]
VmxFile=c:\programme\vmware\vmware workstation\vmpc\vmpc.vmx
VmxSearchStr1=uuid.location
VmxSearchStr2=uuid.bios
VmxSearchStr3=ethernet0.generatedAddress
VmxSearchStr4=displayName

[DEBUG]
DBLVL=0
```

12.6.2 *ip2name.ini (Endgerät)*

```
[MODE]
GetNameMode=ip
FallbackMode=1

[SOURCE]
SysprepTpl=c:\sysprep\sysprep2.inf
SysprepSrchStr=ComputerName=

[TARGET]
Sysprep=c:\sysprep\sysprep.inf
Net=C
HostMask=11111
HostMaskCheck=1
Suffix=PC
CmpNameLength=15

[LOCAL]
LocalInputFile=c:\sysprep\ip2name.txt

[HTTP]
HTTPHost=www.ip2name.org/
HTTPurl=ip2name.txt

[FTP]
FTPHost=ftp.ip2name.org/
FTPurl=ip2name.txt

[MAC]
MacInputFile=c:\sysprep\input.txt
MacFileSearchString=Physikalische Adresse . . . . . . :
MACSize=17

[VM]
VmxFile=c:\programme\vmware\vmware workstation\vmpc\vmpc.vmx
VmxSearchStr1=uuid.location
VmxSearchStr2=uuid.bios
VmxSearchStr3=ethernet0.generatedAddress
VmxSearchStr4=displayName

[DEBUG]
DBLVL=0
```

12.7 Perl-Script *ip2name.pl*

```
#!/perl/bin/perl
use strict;
use Config::IniFiles;
use Sys::Info::OS;
use Data::Dumper; #only needed for full function of Sys::Info::OS
use String::Util ':all';
use LWP::Simple;
use vars '$IniFileName', '$cfg', '$DBLVL', '$DBSGN', '$Nml', '$Mode',
'$FallbackMode', '$SysprepTpl', '$Sysprep', '$SysprepSrchStr', '$Net',
'$Suffix', '$CmpNameLength', '$HostMask', '$HostMaskCheck', '$LocalInputFile',
'$MacInputFile', '$MacFileSrchStr', '$HTTPHost', '$HTTPurl', '$Oktett3',
'$IpAdr', '$CmpName', '$Buffer', '$BufferSize', '$Verbose', '$MAC', '$MACSize',
'$VmxFile', '$VmxFl', '$VmxSearchStr1', '$VmxSearchStr2', '$VmxSearchStr3',
'$VmxSearchStr4';
#
# ip2name.pl
# function: make 'ComputerName' and write into file sysprep.inf
# in detail: 1. get ip-adress of host (DHCP must be enabled)
#            2. read template file sysprep2.inf (default)
#            3. if mode=ip (default)
#            3a. get standard 'HostMask' for Class A,B,C
#                Class A=111111111, Class B=111111 Class C=111
#            3b. or define 'own' HostMask from 111111111...111
#            3c. get suffix-part of Computername (default=PC)
#            3d. make Hostname with part1(Suffix) and part2 (get by HostMask)
#            3e. example1: ip=172.16.7.15 ;Class B; HostMask=111111; Suffix=PC
#                          => Computername=PC007015
#               example2: ip=172.16.7.15 ;Class B; own HostMask=11111;
Suffix=CMP
#                          => Computername=CMP07015
#            4. if mode=ip (http or local)
#            4a. searching for local MAC-Adress in input.txt
#            4b. if mode=local, searching for local MAC-Adress and computername
in ip2name.txt
#            4c. if mode=http, download ip2name.txt and searching for local MAC-
Adress and computername in ip2name.txt
#            5. write new Computername in file sysprep.inf
#
my $Version = "0.4-21.01.2008";
my $Author = "Helmut Thomas";
my $Script = "ip2name";
$IniFileName = "c:\\sysprep\\".$Script.".ini";
$Nml = "...";

#
# main
#
#check command line options
$ARGV[0] =~ tr/A-Z/a-z/;
$ARGV[2] =~ tr/A-Z/a-z/;
while ($_ = $ARGV[0])
      {my $VmMAC = $ARGV[1];
      $VmxFl = $ARGV[2];
      shift;
      last if /^^--$/;
      if (/^-h(.*)/) {&Help}
      elsif (/^-makeini$/) {&MakeIni}
      # not implemented yet
      #elsif (/^-test$/) {&Test}
      elsif (/^-v$/) {$Verbose = "true"}
      elsif (/^-vm$/) {&VirtualMode ($VmMAC, $VmxFl)}

      else {print (":-( ignoring unknown parameter $_, use -? to show options.
```

```
:-)\n")};
       }

if ($Verbose)
  {print "\n";
   print "***********************************************\n";
   print "***\n";
   print "*    $0 is running!\n";
   print "*    Version: $Version\n";
   print "*    Author: $Author\n";
   print "*    Configuration = $IniFileName\n";
   print "*    runnin on Perl $]\n";
   if ($ARGV[0]) {print "*    arguments = $ARGV[0]\n"};
   print "***\n";
   print "***********************************************\n";
  }

#initialize vars
&ReadIni ("");
&Init ("");
if ($Verbose) {print "***********************************************\n"};
#print returned ip-adress
print ($Nml, "IP-Adress='", $IpAdr = &GetIp, "'\n");
if ($Mode ne "ip")
   {print ($Nml, "MAC-Adress='", $MAC = &GetMac($MacFileSrchStr, , $BufferSize,
$MACSize), "'\n");
    #read http into buffer
    eval { $Buffer = &GetFile($LocalInputFile, $BufferSize, $Mode)};
         if ($@ ne "")
            {print ("\nFailure while reading buffer, $@\n");
             if ($FallbackMode == 1)
                {$Mode = "ip";
                 print ("=> working in fallback-mode, changed mode='",
$Mode,"'\n");
                }
             else {exit};
            }
         else {$CmpName = &FindMac($Buffer, $MAC);};
   };
  if ($Mode eq "ip")
     {#get the last 3 oktetts, will be filled to 3 each
      $Oktett3 = &GetOktett;
      #set the length of the hostmask, will be from 3...9
      #111=3, 111111111=9
      $HostMask = &SetHostMask($HostMask);
      #build computername
      $CmpName = &MakeName($Suffix, $Oktett3, $HostMask, $CmpNameLength);
     };
  if ($CmpName eq "") {warn ("!!! Computername='$CmpName' ist empty !!!\n")}
  else {print ($Nml, "Computername='", $CmpName, "'\n")};

#read sysprep2.inf into buffer
eval { $Buffer = &GetFile($SysprepTpl, $BufferSize)};
if ($@ ne "")
   {print ("\nFailure while reading buffer, $@\n");
    exit;}
#modify buffer, if not empty
if (hascontent $Buffer)
   {$Buffer = &ModifyBuffer($Buffer, $SysprepSrchStr, $CmpName);
    if ($Buffer ne "false") {if ($DBLVL == 0, $Verbose) {print ("=>
successfull\n")};}
     else {print ("=> Error in module \$ModifyBuffer!\n");}
    }
else
   {print ("Error: Buffer is empty, maybe $SysprepTpl to?!?\n Exiting...");
    exit};
#write modified buffer
my $Add = ("[$Script]\n"."$CmpName=written by $Script $Version
```

```
(time='".&GetTimeStamp."')\n");
eval {&WriteFile($Sysprep, $Buffer, $Add)};
if ($@ ne "")
  {print ("\nFailure while writing $Sysprep, $@\n");
   print ("!!! $0 not finished sucessfully!!!\n")}
else
   {print ("==> $0 finished!\n");}
exit;

#start sub programs
#

# name of the configuration file (default="$Script.ini")
#
sub ReadIni
{my $tst = $_[0];
 if (!$tst) {print ($Nml,"reading $IniFileName => ")};
 eval {open (IniFileName,"<$IniFileName") || die "could not open $IniFileName.\n
Error: $!";
      close IniFileName};
 if ($@ ne "")
    {if (!$tst)
        {print ("Failure, $@");
        };
    }
   else {if (!$tst) {print("success\n")};
         #read ini-file
         $cfg = Config::IniFiles->new(-file => "$IniFileName");
         # set debug level
         $DBLVL = $cfg->val('DEBUG','DBLVL');
          if (!hascontent $DBLVL) {$DBLVL=0}
         };

  if ($@ ne "")
        {
         $IniFileName ="$Script.ini";
         print ("\n=> try to open '$Script.ini' in current directory => ");
         eval {open (IniFileName,"<$IniFileName") || die "could not open
$IniFileName.\n Error: $!";
               close IniFileName};
         if ($@ ne "")
            {if (!$tst)
                {print ("Failure, $@");
                 print ("\n==> Fallback-Mode, using internal default values:\n")
                };
             $IniFileName ="";
             #force verbose mode
             if (!$tst) {$Verbose = "true"};
             return "";
            }
         else
           {if (!$tst) {print("success\n")};
            $cfg = Config::IniFiles->new(-file => "$IniFileName");
            $DBLVL = $cfg->val('DEBUG','DBLVL');
            if (!hascontent $DBLVL) {$DBLVL=0}
           };
        };

# if ($@ eq "")
 if ($DBLVL > 0)
  {print "*** debug level=$DBLVL\n";
   if ($DBLVL == 1) {$DBSGN = ("\n", "+ ")}
   elsif ($DBLVL == 2) {$DBSGN = ("\n", "++ ")}
   elsif ($DBLVL > 2) {$DBSGN = ("\n", "+++ ")}
   print "*** debug sign=$DBSGN\n";
   print "**********************************************\n";
  }
  else {$DBSGN = ""};
```

```
  return "true";
#end ReadIni
}

# define and initialize vars
#
sub Init
{ my $message = $Nml."value not found => using default ";
  if ($IniFileName)
     {$Mode = $cfg->val('MODE','GetNameMode');
      if (!hascontent $Mode)
         {$Mode = "ip";
          print($message, "GetNameMode=", $Mode, "\n")
         }
      else {$Mode =~ tr/A-Z/a-z/;
            if ($Mode =~ m/(ip|local|http)/)
               {if ($DBLVL > 1) {print ($DBSGN, " Mode '", $Mode, "' o.k.\n")};}
            else {print ("Mode '", $Mode, "' n.o.k.");
                  $Mode = "ip";
                  print ("=> set to default mode='", $Mode,"'\n")};
           };
     $FallbackMode = $cfg->val('MODE','FallbackMode');
      if (!hascontent $FallbackMode)
         {$FallbackMode = 1;
          print($message, "FallbackMode=", $FallbackMode, "\n")
         };
       if ($Mode ne "ip")
          {$LocalInputFile = $cfg->val('LOCAL','LocalInputFile');
           if (!hascontent $LocalInputFile)
              {$LocalInputFile = "c:\\sysprep\\ip2name.txt";
              print($message, "LocalInputFile=", $LocalInputFile, "\n");
          };
       $MacInputFile = $cfg->val('MAC','MacInputFile');
       if (!hascontent $MacInputFile)
          {$MacInputFile = "c:\\sysprep\\input.txt";
           print($message, "MacInputFile=", $MacInputFile, "\n");
          };
       $MacFileSrchStr = $cfg->val('MAC','MacFileSearchString');
       if (!hascontent $MacFileSrchStr)
          {$MacFileSrchStr = "Physikalische Adresse . . . . . . :";
           print($message, "MacFileSearchString=", $MacFileSrchStr, "\n");
          };
      };
       $MACSize = $cfg->val('MAC','MACSize');
       if (!hascontent $MACSize)
          {$MACSize = 17;
           print($message, "MACSize=", $MACSize, "\n");
          };
     if ($Mode eq "http")
        {$HTTPHost = $cfg->val('HTTP','HTTPHost');
         if (!hascontent $HTTPHost)
            {$HTTPHost = "www.ip2name.org/";
             print($message, "HTTPHost=", $HTTPHost, "\n");
            };
         $HTTPurl = $cfg->val('HTTP','HTTPurl');
         if (!hascontent $HTTPurl)
            {$HTTPurl = "ip2name.txt";
             print($message, "HTTPurl=", $HTTPurl, "\n");
            };
        };
   $SysprepTpl = $cfg->val('SOURCE','SysprepTpl');
    if (!hascontent $SysprepTpl)
      {$SysprepTpl = "c:\\sysprep\\sysprep2.inf";
       print($message, "SysprepTpl=", $SysprepTpl, "\n")
      };
    $SysprepTpl =~ tr/A-Z/a-z/;
    $SysprepSrchStr = $cfg->val('SOURCE','SysprepSrchStr');
    if (!hascontent $SysprepSrchStr)
```

```
        {$SysprepSrchStr = "ComputerName=";
        print($message, "SysprepSrchStr=", $SysprepSrchStr, "\n")
        };
      $Sysprep = $cfg->val('TARGET','Sysprep');
      if (!hascontent $Sysprep)
        {$Sysprep = "c:\\sysprep\\sysprep.inf";
        print($message, "Sysprep=", $Sysprep, "\n")
        };
      $Sysprep =~ tr/A-Z/a-z/;
      $Net = $cfg->val('TARGET','Net');
      $Net =~ tr/A-Z/a-z/;
      if (!hascontent $Net)
        {$Net = "c";
        print($message, "Net=", $Net, "\n")
        }
        else {if ($DBLVL >1) {print ($DBSGN, "checking \$Net=$Net\n")};
              if ($Net =~ m/^[a,b,c]{1}$/)
                 {if ($DBLVL >1) {print ($DBSGN, "Net '", $Net, "' is
correct\n")};
                 }
              else {print ($Nml, "Net '", $Net, "' not correct => set to default
'");
                    print ($Net = "c", "'\n")};
             };
      #^ 'start at begin'
      #\d 'only digits'
      #(1+) 'at least one or more 1'
      #[2..9]{0} 'no 2,3,4,5,6,7,8,9'
      $HostMask = $cfg->val('TARGET','HostMask');
      if (hascontent $HostMask)
         {if ($HostMask =~ m/^\d(1+)([2..9]{0})$/)
             {if ($DBLVL >1) {print ($DBSGN, "HostMask ok '", $HostMask, "'\n")};
             }
          else
             {print ("Hostmask '", $HostMask, "' not correct => set to default
'");
              print ($HostMask = "111", "'\n");
             }
          };
      $HostMaskCheck = $cfg->val('TARGET','HostMaskCheck');
      if (!hascontent $HostMaskCheck)
        {$HostMaskCheck = 1; #pattern matching ToDo, syntax check ?
        print($message, "HostMaskCheck=", $HostMaskCheck, "\n")
        };
      $Suffix = $cfg->val('TARGET','Suffix');
      if (!hascontent $Suffix)
        {$Suffix = "PC";
        print($message, "Suffix=", $Suffix, "\n")
        };
      $CmpNameLength = $cfg->val('TARGET','CmpNameLength');
      if (!hascontent $CmpNameLength) #pattern matching, syntax check ToDo ?
        {$CmpNameLength = 15;
        print($message, "CmpNameLength=", $CmpNameLength, "\n")
        };
      if (hascontent $VmxFl)
         {if ($DBLVL >1) {print ($DBSGN, "\$VmxFl='",$VmxFl,"'\n")};
          $VmxFile = $VmxFl;
          print ($Nml, "use command-line parameter VmxFile='", $VmxFile, "'\n");
         }
         else {$VmxFile = $cfg->val('VM', 'VmxFileServer')};
      if ($DBLVL >1) {print ($DBSGN, "\$VmxFile='",$VmxFile,"'\n")};
      $VmxSearchStr1 = $cfg->val('VM', 'VmxSearchStr1');
      $VmxSearchStr2 = $cfg->val('VM', 'VmxSearchStr2');
      $VmxSearchStr3 = $cfg->val('VM', 'VmxSearchStr3');
      $VmxSearchStr4 = $cfg->val('VM', 'VmxSearchStr4');
     }
     else
      {$Mode = "ip";
```

```
      $FallbackMode = 1;
      $SysprepTpl = 'c:\sysprep\sysprep2.inf';
      $SysprepSrchStr = 'ComputerName=';
      $Sysprep = 'c:\sysprep\sysprep.inf';
      $Net = 'c';
      $HostMask = "";
      $HostMaskCheck = 1;
      $Suffix = 'PC';
      $CmpNameLength = 15;
      $LocalInputFile = 'c:\sysprep\ip2name.txt';
      $MacInputFile = 'c:\sysprep\input.txt';
      $MacFileSrchStr = 'Physikalische Adresse . . . . . . :';
      $MACSize = 17;
      $HTTPHost = 'www.ip2name.org/';
      $HTTPurl = 'ip2name.txt';
      $VmxFile = 'c:\programme\vmware\vmware server\vmpc\vmpc.vmx';
      $VmxSearchStr1 = 'uuid.location';
      $VmxSearchStr2 = 'uuid.bios';
      $VmxSearchStr3 = 'ethernet0.generatedAddress';
      $VmxSearchStr4 = 'displayName';
    };

 if ($DBLVL > 0 or $Verbose)
  {print ("***********************************************\n");
   print ("* Mode='$Mode'\n");
   print ("* FallbackMode='$FallbackMode'\n");
   print ("* SysprepTpl='$SysprepTpl'\n");
   print ("* SysprepSrchStr='$SysprepSrchStr'\n");
   print ("* Sysprep='$Sysprep'\n");
   print ("* Net='$Net'\n");
   print ("* HostMask='$HostMask'\n");
   print ("* HostMaskCheck='");
   if ($HostMaskCheck == 1)
     {print ("$HostMaskCheck' \(enabled\)\n")}
     else
      {print ("$HostMaskCheck' \(disabled\)\n")};
   print ("* Suffix='$Suffix'\n");
   print ("* CmpNameLength='$CmpNameLength'\n");
   if ($Mode ne "ip")
      {print ("* LocalInputFile='$LocalInputFile'\n");
       print ("* MacInputFile='$MacInputFile'\n");
       print ("* MacFileSearchString='$MacFileSrchStr'\n");
       print ("* MACSize='$MACSize'\n");};
   if ($Mode eq "http")
      {print ("* HTTPHost='$HTTPHost'\n");
       print ("* HTTPurl='$HTTPurl'\n");};
  }

 $IpAdr = "";
 $CmpName = "";
 $Buffer = "";

 # define internal constant
 $BufferSize = 4000;
 if ($DBLVL > 1) {print ($DBSGN, "BufferSize='", $BufferSize, "'\n")};
 }

 sub GetIp
 {  my $SubName = "GetIp";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $Ip;
    my $tst = $_[0];
    my $OS = Sys::Info::OS->new(my %options);
    warn "Collected information can be incomplete\n" if $OS->is_unknown;
    $Ip = $OS->ip;
    warn "\a!!! Only Loopback detected! Check your network settings!!!\n" if
 (($Ip eq "127.0.0.1") and (!$tst));
    #first oktett 1-254.=
```

```
   # ([1-9]{1}\d?)|(1{1}\d{0,2})|(2{1}[0-4]?[0-9]?)|(2{1}5?[0-4]?))\.
   #other oktetts 0-254.=
   # (\d{0,2})|(1{1}\d{0,2})|(2{1}[0-4]?[0-9]?)|(2{1}5?[0-4]?))\.
   # not valid $Ip = "111.222.333.444";
   if ($Ip =~ m/^(([1-9]{1}\d?)|(1{1}\d{0,2})|(2{1}[0-4]?[0-9]?)|(2{1}5?[0-
4]?))\.((\d{0,2})|(1{1}\d{0,2})|(2{1}[0-4]?[0-9]?)|(2{1}5?[0-
4]?))\.((\d{0,2})|(1{1}\d{0,2})|(2{1}[0-4]?[0-9]?)|(2{1}5?[0-
4]?))\.((\d{0,2})|(1{1}\d{0,2})|(2{1}[0-4]?[0-9]?)|(2{1}5?[0-4]?))$/)
      {if ($DBLVL > 1) {print ($DBSGN, "Ip $Ip is valid.\n")}}
    else
        {print ($Nml, "\aIp $Ip is not (!) valid ");
         $Ip = "127.0.0.1";
         print ($Nml, "=> set it to loopback '", $Ip, "'\n");
         }
   if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
   return $Ip;
}

sub GetMac
{   my $SubName = "GetMac";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $SrchStr = $_[0];
    my $BffrSz = $_[1];
    my $Offset = $_[2];
    my $Bff;
    if ($Verbose) {print ($Nml, "try to open '$MacInputFile'\n")};
    open (MacInputFile,"<$MacInputFile") || die "could not open $MacInputFile.\n
Error:$!";
    my $BufferLgth = read(MacInputFile, $Bff, $BffrSz);
    close MacInputFile;
    if ($DBLVL > 1) {print ($DBSGN, "BufferSize=$BufferLgth\n")};
    if ($DBLVL > 2) {print ($DBSGN, "Content Buffer=\n$Bff\n")};
    my $SrchStrPos = index($Bff, $MacFileSrchStr);
    if ($SrchStrPos != -1)
       {if ($Verbose){print ($Nml, "tag '$MacFileSrchStr' found at position
$SrchStrPos\n")}}
    else {if ($Verbose) {print ($Nml, "'$MacFileSrchStr' not found")};
         return "false"};
    if ($DBLVL > 1) {print ($DBSGN, "SrchStrPos=$SrchStrPos\n")};
    my $Mac = substr($Bff, ($SrchStrPos + length($MacFileSrchStr) + 1),
$Offset);
    if ($DBLVL > 1) {print ($DBSGN, "\$Mac ='", $Mac, "'\n")};
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $Mac;
}

sub FindMac
{   my $SubName = "FindMac";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $Bffr = $_[0];
    my $Mac = $_[1];
    my $CmpNm = "";
    my $OneChar = "";
    print ($Nml, "searching for MAC '", $Mac,"'\n");
    my $Pos = (index ($Bffr, $Mac)) + length ($Mac);
    if ($Pos == (length ($Mac) -1)) {warn ("\a!!! $Mac not found !!!\n"); return
""};
    if ($DBLVL > 2) { print ($DBSGN, "\$Pos='", $Pos, "'
length(\$Mac)='",length($Mac),"\n")};
    print ($Nml, "Found '", $Mac, "' at position ", $Pos,"\n");
    $OneChar = substr ($Bffr, ++$Pos, 1);
    if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
    while ($OneChar ne ";")
          {$CmpNm .= $OneChar;
           $OneChar = substr ($Bffr, ++$Pos, 1);
           if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
          }
    $CmpNm = nospace ($CmpNm); #remove all whitespace [String::Util]
```

```
    if ($DBLVL > 0) {print ($DBSGN, "\$CmpNm ='", $CmpNm, "'\n")};
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $CmpNm;
}

sub GetOktett
{   my $SubName = "GetOktett";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $OneChar;
    my ($FstOktett, $MidOktett, $LstOktett);
    my $Pos = index ($IpAdr, ".");
    if ($DBLVL > 1)
      {print ($DBSGN, "first '.' found at Pos $Pos\n")};
    if ($DBLVL > 1) { print ($DBSGN, "IpAdr='", $IpAdr, "'\n")};
    $OneChar = substr ($IpAdr, ++$Pos, 1);
    if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
    while ($OneChar ne ".")
    {   $FstOktett .= $OneChar;
        $OneChar = substr ($IpAdr, ++$Pos, 1);
        if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
    }
    if ($DBLVL > 0) {print ($DBSGN, "FstOktett=", $FstOktett, "\n")};

    $OneChar = substr ($IpAdr, ++$Pos, 1);
    if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
    while ($OneChar ne ".")
    {   $MidOktett .= $OneChar;
        $OneChar = substr ($IpAdr, ++$Pos, 1);
        if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
    }
    if ($DBLVL > 0) {print ($DBSGN, "MidOktett=", $MidOktett, "\n")};

    $OneChar = substr ($IpAdr, ++$Pos, 1);
    if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
    while ($OneChar ne "")
    {   $LstOktett .= $OneChar;
        $OneChar = substr ($IpAdr, ++$Pos, 1);
        if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
    }
    if ($DBLVL > 0) {print ($DBSGN, "LstOktett=", $LstOktett, "\n")};

    $Oktett3 = &FillOktett ($FstOktett).&FillOktett ($MidOktett).&FillOktett
($LstOktett);
    if ($DBLVL > 1) { print ($DBSGN, "Oktett3=", $Oktett3, "\n")};
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $Oktett3;
}

sub FillOktett
{   my $SubName = "FillOktett";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $Oktett = $_[0];
    if ($DBLVL > 1) {print ($DBSGN, "FillOktett \$Oktett = $Oktett\n")};
    # fill oktett to length of 3
    if (length($Oktett)== 1) {$Oktett = "00" . $Oktett;}
    elsif (length($Oktett)== 2) {$Oktett = "0" . $Oktett;;}
    elsif (length($Oktett)> 3) {print "Error in Oktett Oktett\n";}
    if ($DBLVL > 0) {print ($DBSGN, "Oktett (modified)=", $Oktett, "\n")};
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $Oktett;
}

sub GetHostMask
{   my $SubName = "GetHostMask";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $HstMsk = $_[0];
    my $HstMskMin = 3;
    my $HstMskMax = 9;
```

```
    warn ("\n\aWARNING: Parameter 'HostMask=$HstMsk' used, ignoring Parameter
'Net'!\n");
    if ($DBLVL > 1)
      {print ($DBSGN, "HostMask (length) before verify=", length($HstMsk),
"\n")};
    if (length($HstMsk) < $HstMskMin and $HostMaskCheck == 1)
       { print ("HostMask too short, length must be ",$HstMskMin,"-",
$HstMskMax, "\n");
         print ("HostMask length set to ", $HstMskMin,", using '111'\n");
         $HstMsk = $HstMskMin;
       }
    else {$HstMsk = length($HstMsk);
          if ($HstMsk > $HstMskMax)
             {print ("HostMask to long, length must be ",$HstMskMin,"-",
$HstMskMax, "\n");
              print ("HostMask length set to ", $HstMskMax,", using
'111111111'\n");
              $HstMsk = $HstMskMax};
          if ($DBLVL > 1) {print ($DBSGN, "HostMask ='", $HstMsk, "'\n")};
          }
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $HstMsk;
}

sub SetHostMask
{   my $SubName = "SetHostMask";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $HstMsk = $_[0];
    if (hascontent $HstMsk)
      {$HstMsk = &GetHostMask($HstMsk);
        if ($DBLVL > 1) {print ($DBSGN, "returned HostMask=$HstMsk\n")};
      }
    elsif ($Net eq "a") {$HstMsk = 9}
    elsif ($Net eq "b") {$HstMsk = 6}
    elsif ($Net eq "c") {$HstMsk = 3}

    print ($Nml, "using HostMask='");
    my $i = 0;
    for ($i=0; $i<$HstMsk; $i++) {print ("1")};
    print ("'\n");
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $HstMsk;
}

sub MakeName
{   my $SubName = "MakeName";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName with
HostMask=$HostMask\n")};
    my $Pr = $_[0];
    my $Okt = $_[1];
    my $HstMsk = $_[2];
    my $CmpNmLgth = $_[3];
    my $CmpNm = $Pr . substr($Okt, length($Okt)-$HstMsk, $HstMsk);
    print ($Nml, "ComputerName='$CmpNm'\n");
    if (length($CmpNm) > $CmpNmLgth)
       {print ($Nml, "Computername longer than $CmpNmLgth, crunched to:",
               $CmpNm = substr($CmpNm, -$CmpNmLgth), "\n");
       }
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $CmpNm;
}

sub GetFile
{   my $SubName = "GetFile";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $GetFile = $_[0];
    my $BffrSz = $_[1];
    my $GetFileMode = $_[2];
```

```
    my $Bff = "";
    my $BufferLgth;
    if ($Verbose) {print ($Nml, "try to open '$GetFile'\n")};
    if ($GetFileMode eq "http")
       {if ($DBLVL > 0) {print ($DBSGN, "GetFile=$GetFile from 'http://",
$HTTPHost.$HTTPurl,"'\n")};
        my $status = getstore("http://$HTTPHost$HTTPurl", $GetFile);  #
[LWP::Simple]
        if (is_success($status)) # [LWP::Simple]
           {print ($Nml, "download from 'http://", $HTTPHost.$HTTPurl,"'
o.k.\n");
            print ($Nml, "making local '", $GetFile, "' o.k.\n");
           }
          else {print ($GetFile, "!!! n.o.k. !!!\n");};
       };
    open (GetFile,"<$GetFile") || die "could not open $GetFile.\n Error: $!";
    $BufferLgth = read(GetFile, $Bff, $BffrSz);
    close GetFile;
    if ($DBLVL > 1) {print ($DBSGN, "BufferSize=$BufferLgth\n")};
    if ($DBLVL > 2) {print ($DBSGN, "Content Buffer=\n$Bff\n")};
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $Bff;
}

sub ModifyBuffer
{   my $SubName = "ModifyBuffer";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $Bffr = $_[0];
    my $SsprpSrchStr = $_[1];
    my $CmpNm = $_[2];
    my $SrchStrPos = index($Bffr, $SsprpSrchStr);
    if ($SrchStrPos != -1)
       {if ($Verbose){print ($Nml, "tag '$SsprpSrchStr' found at position
$SrchStrPos\n")}}
    else {if ($Verbose) {print ($Nml, "'$SsprpSrchStr' not found")};
         return "false"};
    if ($DBLVL > 1) {print ($DBSGN, "SrchStrPos=$SrchStrPos\n")};
    if ($DBLVL > 1) {print ($DBSGN, "CmpName=$CmpNm\n")};
    if ($Verbose) {print ($Nml, "modified '$SsprpSrchStr$CmpNm' ")};
    if ($DBLVL > 1) {print ("\n")};
    substr($Bffr, $SrchStrPos+length($SsprpSrchStr), 0) = $CmpNm;
    if ($DBLVL > 2) {print ($DBSGN, "Content modified Buffer=\n$Bffr\n")};
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $Bffr;
}

sub ModifyBufferVM
{   my $SubName = "ModifyBufferVM";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $Bffr = $_[0];
    my $SrchStr = $_[1];
    my $Replace = $_[2];
    my $RplMode = $_[3];
    my $OneChar = "";
    my $SrchStrPos = index($Bffr, $SrchStr);
    if ($SrchStrPos != -1)
       {if ($Verbose){print ($Nml, "tag '$SrchStr' found at position
$SrchStrPos\n")}}
    else {if ($Verbose) {print ($Nml, "'$SrchStr' not found")};
         return "false"};
    if ($DBLVL > 1) {print ($DBSGN, "SrchStrPos=$SrchStrPos\n")};
    if ($DBLVL > 1) {print ($DBSGN, "Replace=$Replace\n")};
    if ($DBLVL > 1) {print ("\n")};
    my $Pos = (index ($Bffr, $SrchStr)) + length ($SrchStr);
    if ($DBLVL > 1) { print ($DBSGN, "Found '", $SrchStr, "' at position ",
$Pos,"\n")};
    $OneChar = substr ($Bffr, ++$Pos, 1);
    if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar, "'\n")};
```

```
    # searching for 1. sign '"', if 'RplMode=unpair'
    # searching for 2. sign '"', if 'RplMode=pair'
    my $OneOrTwo;
    if (($RplMode == 2) or ($RplMode == 4)) {$OneOrTwo = 2} else {$OneOrTwo =
1};
    my $i = 0;
    for ($i=0; $i<$OneOrTwo; $i++)
    {   while ($OneChar ne "\"")
        {      $Pos = $Pos + 1;
               $OneChar = substr ($Bffr, $Pos, 1);
               if ($DBLVL > 1) { print ($DBSGN, "Found char='", $OneChar,
"'\n")};
        };
        $Pos = $Pos + 1;
        $OneChar = substr ($Bffr, $Pos, 1);
    };
    if ($DBLVL > 1) { print ($DBSGN, "sign '\"' found at ", $Pos, "'\n")};
    if ($RplMode == 1) {substr($Bffr, $Pos, length ($Replace)) = $Replace};
    if ($RplMode == 2) {substr($Bffr, ($Pos - length ($Replace)- 1), length
($Replace)) = $Replace};
    if ($RplMode == 3) {substr($Bffr, $Pos, 0) = $Replace};
    if ($RplMode == 4) {substr($Bffr, $Pos -1, 0) = $Replace};
    if ($DBLVL > 2) {print ($DBSGN, "Content modified Buffer=\n$Bffr\n")};
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $Bffr;
}

sub WriteFile
{   my $SubName = "WriteFile";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    my $Ssprp = $_[0];
    my $Bffr = $_[1];
    my $Ad = $_[2];
    if ($Verbose) {print ($Nml, "try to open and write '$Ssprp'\n")};
    open (Ssprp,"+>$Ssprp") || die "could not open $Ssprp.\n Error:$!";
    print Ssprp "$Bffr\n";
    if ($Ad) {print Ssprp $Ad};
    close Ssprp;
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
}

sub MakeIni
{   my $SubName = "MakeIni";
    my $FileName = $Script.".ini.default";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    print ("making file '$FileName' and exit.\n");
    open (FileName,"+>$FileName") || die "File $FileName could not be opened.
Error$!";
    print FileName "#\n";
    print FileName "#generated by $0 $Version (time='", &GetTimeStamp, "')\n";
    print FileName "\n";
    print FileName "[MODE]\n";
    print FileName "# GetNameMode: [ip|local|http|ftp]\n";
    print FileName "# GetNameMode defines the method for building the name\n";
    print FileName "# ip: builds the name with the gives ip-adresse\n";
    print FileName "# uses [SOURCE] and [TARGET]\n";
    print FileName "# local: looking for input-file\n";
    print FileName "# uses [SOURCE], [TARGET], [LOCAL] and [MAC]\n";
    print FileName "# http: looking for input-file at url\n";
    print FileName "# uses [SOURCE], [TARGET], [LOCAL], [MAC] and [HTTP]\n";
    print FileName "# ftp: not implemented yet\n";
    print FileName "# GetNameMode=ip (default)\n";
    print FileName "GetNameMode=ip\n";
    print FileName "\n";
    print FileName "# FallbackMode: enable/disable fallback to ip-mode, if\n";
    print FileName "# desired GetNameMode failed\n";
    print FileName "# enable=1/disable=0\n";
    print FileName "FallbackMode=1\n";
```

```
    print FileName "\n";
    print FileName "[SOURCE]\n";
    print FileName "# SysprepTpl: templ.File for building sysprep.inf\n";
    print FileName "# SysprepTpl=c:\\sysprep\\sysprep2.inf (default)\n";
    print FileName "SysprepTpl=c:\\sysprep\\sysprep2.inf\n";
    print FileName "\n";
    print FileName "# SysprepSrchStr: string to search for in sysprep2.inf\n";
    print FileName "# SysprepSrchStr=ComputerName= (default)\n";
    print FileName "SysprepSrchStr=ComputerName=\n";
    print FileName "\n";
    print FileName "[TARGET]\n";
    print FileName "# Sysprep: must named sysprep.inf and placed in
c:\\sysprep\n";
    print FileName "# Sysprep=c:\\sysprep\\sysprep.inf (default)\n";
    print FileName "Sysprep=c:\\sysprep\\sysprep.inf\n";
    print FileName "\n";
    print FileName "# Net: Class A, B or C\n";
    print FileName "# Net=A results in HostMask=111111111\n";
    print FileName "# Net=B results in HostMask=111111\n";
    print FileName "# Net=C results in HostMask=111\n";
    print FileName "# Net=C (default)\n";
    print FileName "Net=C\n";
    print FileName "\n";
    print FileName "# HostMask must read from right to left\n";
    print FileName "# indicates the bits used for building hostname\n";
    print FileName "# use of HostMask will override Parameter Net\n";
    print FileName "#\n";
    print FileName "HostMask=\n";
    print FileName "\n";
    print FileName "# HostMaskCheck: enable/disable minimum length of 3 for
HostMask\n";
    print FileName "# enable=1/disable=0\n";
    print FileName "# HostMaskCheck=1\ (default)\n";
    print FileName "HostMaskCheck=1\n";
    print FileName "\n";
    print FileName "# Suffix: first Part of the Computername\n";
    print FileName "# Suffix=PC (default)\n";
    print FileName "Suffix=PC\n";
    print FileName "\n";
    print FileName "# crunches the resulting computername, if to long\n";
    print FileName "# CmpNameLength=15 (default)\n";
    print FileName "CmpNameLength=15\n";
    print FileName "\n";
    print FileName "[LOCAL]\n";
    print FileName "# LocalInputFile=c:\\sysprep\\ip2name.txt (default)\n";
    print FileName "LocalInputFile=c:\\sysprep\\ip2name.txt\n";
    print FileName "\n";
    print FileName "[HTTP]\n";
    print FileName "# HTTPHost=www.ip2name.org (default)\n";
    print FileName "HTTPHost=www.ip2name.org/\n";
    print FileName "\n";
    print FileName "# HTTPurl=ip2name.txt (default)\n";
    print FileName "HTTPurl=ip2name.txt\n";
    print FileName "\n";
    print FileName "[FTP]\n";
    print FileName "# FTPHost=ftp.ip2name.org/ (default)\n";
    print FileName "FTPHost=ftp.ip2name.org/\n";
    print FileName "\n";
    print FileName "# FTPurl=ip2name.txt (default)\n";
    print FileName "FTPurl=ip2name.txt\n";
    print FileName "\n";
    print FileName "[MAC]\n";
    print FileName "# files, needed for MAC-Detection\n";
    print FileName "# input.txt can generated by OS, e.g. 'ipconfig –all
>input.txt'\n";
    print FileName "# MacInputFile=c:\\sysprep\\input.txt (default)\n";
    print FileName "MacInputFile=c:\\sysprep\\input.txt\n";
    print FileName "\n";
```

```
    print FileName "# searching for MacFileSearchString in input.txt\n";
    print FileName "# MacFileSearchString=Physikalische Adresse . . . . . . :
(internal german default)\n";
    print FileName "MacFileSearchString=Physikalische Adresse . . . . . . :\n";
    print FileName "\n";
    print FileName "# MACSize=17 chars, eg. '11:22:33:44:55:66'\n";
    print FileName "# MACSize=17 (default)\n";
    print FileName "MACSize=17\n";
    print FileName "\n";
    print FileName "[VM]\n";
    print FileName "# use for virtual machine deployment only\n";
    print FileName "# VmxSearchStr1=uuid.location (default)\n";
    print FileName "# VmxSearchStr2=uuid.bios (default)\n";
    print FileName "# VmxSearchStr3=ethernet0.generatedAddress (default)\n";
    print FileName "# VmxFile: destination, where to find the vmx-file to be
edited\n";
    print FileName "# VmxFile: command-line parameter will override ini  -
parameter\n";
    print FileName "VmxFile=c:\\programme\\vmware\\vmware
workstation\\vmpc\\vmpc.vmx\n";
    print FileName "VmxSearchStr1=uuid.location\n";
    print FileName "VmxSearchStr2=uuid.bios\n";
    print FileName "VmxSearchStr3=ethernet0.generatedAddress\n";
    print FileName "VmxSearchStr4=displayName\n";
    print FileName "\n";
    print FileName "[DEBUG]\n";
    print FileName "# DBLVL: possible Values:\n";
    print FileName "# 0 = no debug (default)\n";
    print FileName "# 1 = normaler debug\n";
    print FileName "# 2 = strong debug\n";
    print FileName "DBLVL=0\n";
    close FileName;
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    exit;
}

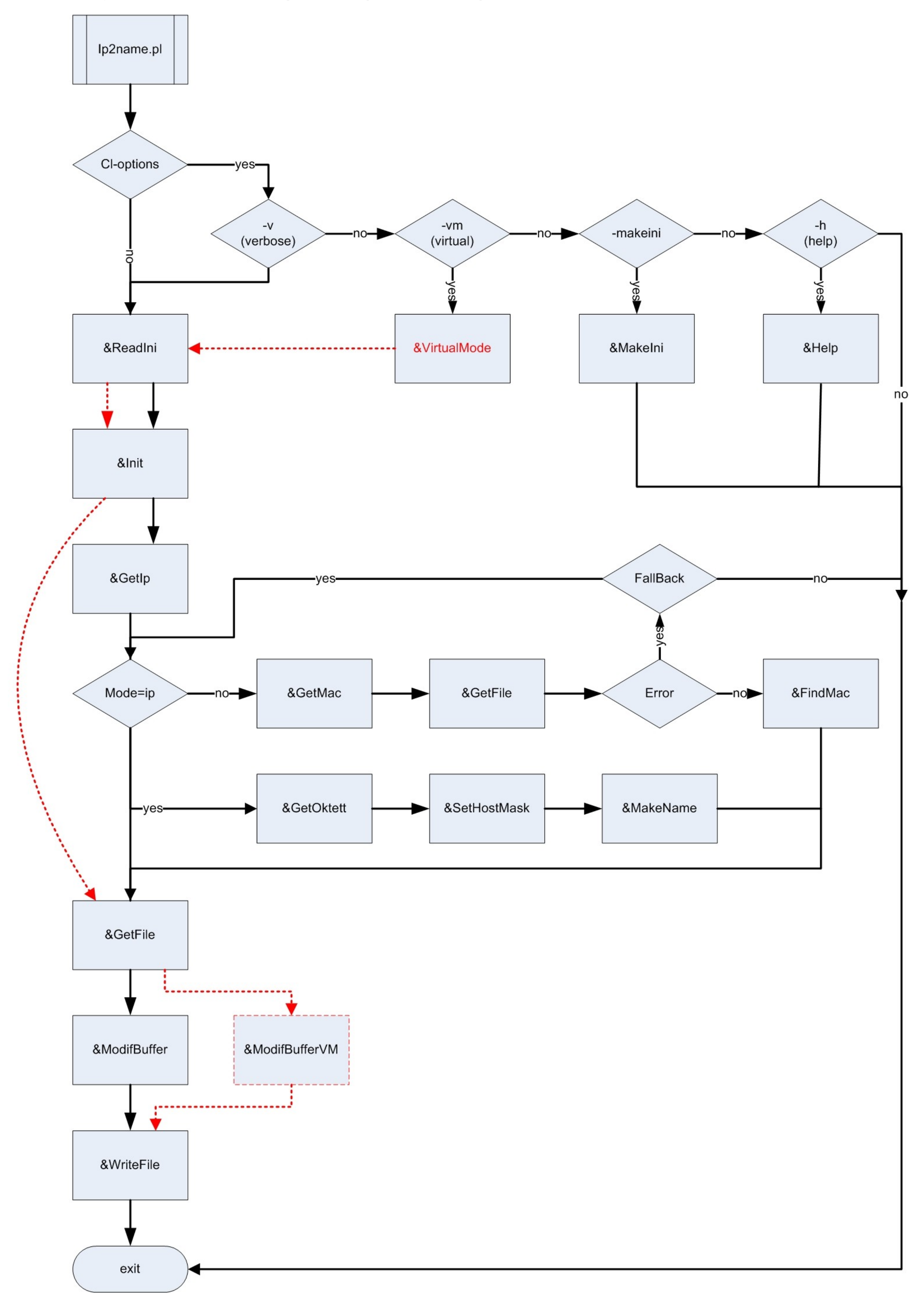
sub Help
{   my $SubName = "Help";
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    print ("\n");
    print ("$0 [-h|-makeini|-test|-v|-vm]\n");
    print ("        -h:       display this help\n");
    print ("        -makeini: generates a default ini-file named '",
$IniFileName, ".default'\n");
    # not implemented yet
    #print ("        -test:    test with parameters read from '", $IniFileName,
"'\n");
    print ("        -v:       verbose, display a little bit more...\n");
    print ("        -vm:      --MAC-Adress [--VMwareVmxFile]\n");
    print ("        --MAC-Adress:    in format '11:22:33:44:55:66'\n");
    print ("        --VMwareVmxFile: complete path to the .vmx-file; if missing
the\n");
    print ("                         value will be read from the ini-file\n");
    print ("\n");
    print ("example: $0 -vm 11:22:33:44:55:66 d:\\vmware\\mymachine.vmx\n");
    print ("\n");
    print ("Prerequisites: (a) installed perl ;-)\n");
    print ("               (b) perl-modules:  Config::IniFiles\n");
    print ("                                  Sys::Info::OS\n");
    print ("                                  Data::Dumper\n");
    print ("                                  String::Util\n");
    print ("                                  LWP::Simple\n");
    print ("               (c) a ini-file named '", $IniFileName,"' or '",
$Script,".ini'\n");
    print ("                   in the same directory\n");
    print ("                   recommended: '", $IniFileName,"'\n");
    print ("               (d) a template file named 'sysprep2.inf' for building
a \n");
```

```
    print ("                    new sysprep.inf\n");
    print ("               (e) if not runnin' in 'ip'-mode, a file 'input.txt',
where \n");
    print ("                    to find the own MAC-Adress of the active
interface,\n");
    print ("                    e.g. generated with 'ipconfig -all >input.txt'
before\n");
    print ("                    calling $0\n");
    print ("               (f) suffient rights in the directory to write
'sysprep.inf'\n");
    print ("               (g) in 'virtual Mode', suffient rights in the
directory to\n");
    print ("                     change the vmx-File in the given VMWare-
directory\n");
    print ("\n");
    print ("<> see README for further details! <>\n");
    print ("<> (c) by $Author 2006-2008\n");
    print ("<> but it's perl and free to use and distribute it :-)\n");
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    exit;
}
#
# begin virtual mode
#
sub VirtualMode
{   my $SubName = "VirtualMode";
    my $Mc = $_[0];
    if ($Mc eq "") {print("=> missing paramter MAC in format
'11:22:33:44:55:66'\n"); &Help;};
    $Mc =~ tr/A-Z/a-z/;
    &ReadIni ("");
    &Init ("");
    if ($DBLVL > 1) {print ($DBSGN, "starting sub $SubName\n")};
    print ($Nml, "script in $SubName started\n");
    my $McTrans = $Mc;
    if (length($Mc) != $MACSize) {print ($Nml, "MAC-Adress '", $Mc, "' has wrong
length!\n"); exit;}
       elsif ($Mc =~ m/^((([0-9,a,b,c,d,e,f]{2}):{1}){5})([0-
9,a,b,c,d,e,f]{2})$/)
             {print ($Nml, "pattern match with MAC '", $Mc, "' ok\n")}
       else  {print ($Nml, "Parameter '", $Mc, "' is not a valid MAC-Adress\n");
exit;};
    if ($DBLVL > 1) {print ($DBSGN, "use parameter MAC=$Mc\n")};
    $Buffer = (&GetFile ($VmxFile, $BufferSize));
    if ($DBLVL > 1) {print ($DBSGN, "\$Buffer =\n", $Buffer, "\n")};
    # transformation 1: '11:22:33:44:55:66' -> '11 22 33 44 55 66'
    if ($DBLVL > 1) {print ($DBSGN, "transformation  = '", $McTrans, "'\n")};
    $McTrans =~ s/:{1}/ /g;
    if ($DBLVL > 1) {print ($DBSGN, "transformation 1= '", $McTrans, "'\n")};
    # transformation 2: '11 22 33 44 55 66' -> '44 55 66'
    $McTrans = substr ($McTrans, -8);
    if ($DBLVL > 1) {print ($DBSGN, "transformation 2= '", $McTrans, "'\n")};
    # ReplaceMode: 1=from begin, left-to-right;
    #              2=from end, right-to-left
    #              3=add (no replace!) from begin, left-to-right
    #              4=add (no replace!) from end, right-to-left
    my $ReplaceMode = 2;
    #modify 'uuid.location' (default)
    $Buffer = (&ModifyBufferVM ($Buffer, $VmxSearchStr1, $McTrans,
$ReplaceMode));
    #modify 'uuid.bios' (default)
    $Buffer = (&ModifyBufferVM ($Buffer, $VmxSearchStr2, $McTrans,
$ReplaceMode));
    my $ReplaceMode = 1;
    #modify 'ethernet0.generatedAddress' (default)
    $Buffer = (&ModifyBufferVM ($Buffer, $VmxSearchStr3, $Mc, $ReplaceMode));
    # transformation 3: '44 55 66' -> '445566' #remove whitespace [String::Util]
```

```
    $McTrans = nospace ($McTrans);
    my $ReplaceMode = 4;
    if ($DBLVL > 1) {print ($DBSGN, "transformation 3= '", $McTrans, "'\n")};
    #modify 'displayName' (default)
    $Buffer = (&ModifyBufferVM ($Buffer, $VmxSearchStr4, $McTrans,
$ReplaceMode));
    if ($DBLVL > 1) {print ($DBSGN, "\$Buffer =\n", $Buffer, "\n")};
    my $Add = "";
    &WriteFile($VmxFile, $Buffer, $Add);
    print ($Nml, "'$VmxFile' changed and written\n");
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    exit;
}

sub GetTimeStamp
{   my $SubName = "GetTimeStamp";
    my ($sek,$min,$std,$mtag,$mon,$jahr,$wtag,$jtag,$isdst) = localtime(time);
    my $date = $mtag.".".($mon+1).".".($jahr+1900);
    my $time = $std.":".$min.":".$sek;
    my $TimeStamp=$date."#".$time;
    if ($DBLVL > 1) {print ("TimeStamp=", $TimeStamp, "\n")};
    if ($DBLVL > 1) {print ($DBSGN, "end sub $SubName\n")};
    return $TimeStamp;
}
```

12.8 Programmablaufplan *ip2name.pl*



12.9 Gebräuchliche Dateiendungen

Dateiendungen sind in der Windows-Welt wesentlich verbreiteter als in der Unix- und Linux-Welt. Bis auf einige Ausnahmen, wie z.B. *conf*, *gz*, *pl und rpm*, ist die Welt unter Unix und Linux frei von Endungen. Auch die Endung *conf* wird nicht wirklich benötigt, sondern dient lediglich der Kennzeichnung, dass es sich hierbei um eine Datei handelt, die Konfigurationsinformationen enthält. Daher finden sich die meisten der hier erwähnten Endungen auch nur in Windows wieder.

Endung	dient wozu
cmd	Batchscripte, die vom Befehlsinterpreter *cmd.exe* ausgeführt werden.
com	Ausführbare Dateien unter Windows.
exe	Ausführbare Dateien unter Windows.
gz	Die übliche Dateiendung für *gzip*-komprimierte Dateien ist heute *.gz*, früher auch *.z*.
idf	Image-Definiton-File; enthält Konfigurationsdefinition für die Softwareauswahl des Thin-Clients mit eLux NG.
inf	Informationsdateien für Installationen von Treibern .
ini	Wird für Initialisierungsdatei benutzt und stellt eine Textdatei dar, die zum Speichern und Einlesen von Programmeinstellungen verwendet wird.
msi	Pakete im Microsoft Installer Format.
pl	Die Dateiendung für Perl-Programme.
pm	Die Dateiendung für Perl-Module und -Pakete.
rpm	Red Hat Package Manager, wird für Archive genutzt, die ein Softwarepaket enthalten.
sys	Windows Systemdateien.
tar	Entstanden aus tape archiver; bezeichnet die Endung von Tarballs.
txt	Dateiendung für reine Textdateien, wird vorwiegend bei DOS und Windows benutzt.
vmdk	Virtual Machine Disk Format; entspricht einem Festplattenlaufwerk in einer virtuellen Maschine.
vmsd	Dies ist eine zentralisierte Datei zum Speichern von Informationen und Metadaten über Momentaufnahmen.
vmsn	Dies ist die Momentaufnahmen-Statusdatei, die den aktiven Zustand einer virtuellen Maschine zum Zeitpunkt der Erstellung einer Momentaufnahme speichert.
vmss	Dies ist die Suspend-State-Datei, die den Status einer angehaltenen virtuellen Maschine speichert.

vmx	Enthält Konfigurationsinformationen zu einer virtuellen Maschine unter VMWare. Die Datei liegt im Ascii-Format vor.
vmxf	Enthält zusätzliche Informationen, wenn die virtuelle Maschine Teil eines Teams von mehreren virtuellen Maschinen ist.

Literatur

AMD95 AMD: *Magic Packet Technology*. - White Paper, http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/20213.pdf, 1995

Ahn07 Ahnert, Sven: *Virtuelle Maschinen mit VMware und Microsoft*. Addison-Wesley-Verlag, 2007

BSD07 The FreeBSD Documentation Project: *FreeBSD Handbook*,

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/jails.html, 2007

CTX07 Citrix: *Value-add to Microsoft Windows Terminal Services*, http://www.citrix.de/produkte/schnellsuche/presentation-server/list/brochures/, 2007

Eli04 UniCon Software GmbH: *Administrationshandbuch eLux® NG Image Administration Service Next Generation (ELIAS NG)*, 2004

eLu06 UniCon Software GmbH: *Administrationshandbuch eLux® NG –Embedded Linux für Thin Clients Next Generation*, 2006

FhG05 Fraunhofer Gesellschaft: *Wirtschaftlichkeitsbetrachtung Thin Client versus Rich Client*, 2005

FIAO06 Fraunhofer Institut für Arbeitswirtschaft und Organisation (IAO): *Trendstudie Open Source Software –Strukturwandel oder Strohfeuer?*, 2006

KB909264 Microsoft: *Support Article - Naming conventions in Active Directory for computers, domains, sites, and OUs*, 2007

KB908784 Microsoft: *Support Article - New users are still limited by the Internet Explorer Enhanced Security Configuration component in Windows Server 2003 after this component is removed*, 2007

LAN07 LANline: *Zwischen Hype und Wirklichkeit*, LANline Ausgabe 4/2007

Q244036 Microsoft: *Technet Article - Description of PXE Interaction Among PXE Client, DHCP, and RIS Server*, 2003/2007

Sco06 UniCon Software GmbH: *Administrationshandbuch Scout Enterprise*, 2006

Sym03 Symantec: *Symantec Ghost Implementierungshandbuch*, 2003

Tho05 Thomas, Helmut: *Betrachtung zum Einsatz von Windows oder Open Source* Kreisausschussvorlage, 2005

MSD04 Microsoft: *Windows Corporate Deployment Tools* - Benutzerhandbuch, 2004

MSR04 Microsoft: *Windows Preinstallation Reference*, 2004

Wal97 Wall, Larry: *Programmieren mit Perl*, O'Reilly Verlag, 1997

Wik07 Wikipedia: *Wikipedia, die freie Enzyklopädie im Internet*, http://de.wikipedia.org/wiki/

XPP02 Microsoft Corporation: *Microsoft Windows XP Professional – Die technische Referenz*, Microsoft Press, 2002

Index

Glossar

ACPI	Das Advanced Configuration and Power Management Interface ist ein offener Industriestandard für Energieverwaltung in Computern.
Active Directory	Ein Verzeichnisdienst der Fa. Microsoft, der seit der Version Windows 2000 verfügbar ist. Dabei ist es möglich über das Protokoll LDAP Informationen aus dem Verzeichnisdienst abzufragen.
Add-on	Ein Add-on ist ein optionales Modul, welches bestehende Hard- oder Software ergänzt oder erweitert.
APIC	Der Advanced Programmable Interrupt Controller sorgt für die Verteilung von Interrupts in Computersystemen.
Backbone	Im vorliegenden Zusammenhang das Kern-Netzwerk, also das Netzwerk mit denen die Server im Serverraum verbunden sind.
BINL	Boot Information Negotiation Layer - Der BINL-Dienst, die Hauptkomponente von Remote Installation Server (RIS), beantwortet PXE-Clientanfragen, validiert mithilfe des Active Directory den Client und übernimmt den Informationsaustausch zwischen Client und Server.
DLL	Dynamic Link Library - bezeichnet allgemein eine Dynamische Bibliothek, meist bezieht sich der Begriff jedoch auf die unter dem Betriebssystem Microsoft Windows verwendete Variante.
Fat-Client	Ein Fat-Client (oder oft Rich-Client genannt) verarbeitet Daten lokal (im Gegensatz zu einem Thin-Client) und bedient sich dabei i. d. R. einer grafischen Oberfläche. Ein Fat-Client besitzt ein vollwertiges Betriebssystem und kann Anwendungssoftware lokal installieren und betreiben.
Hotfix	Fehlerbehebungen oder Sicherheitsupdates, die gezielt einzelne Fehler oder Sicherheitslücken beheben.
ICA	(Independent Computing Architecture) - Die von MetaFrame Presentation Server verwendete Architektur, mit der die Anwendungslogik von der Benutzeroberfläche der Anwendungen getrennt wird. Mit der ICA-Architektur werden nur die Tastatureingaben, Mausklicks und Bildschirmaktualisierungen zwischen dem Client und dem Server im Netzwerk übertragen. Die Anwendungslogik wird zu 100 % auf dem Server ausgeführt.

IDE	Integrated Disk Elektronik; Controllerhardware auf dem Massenspeicher (Festplatte, CD/DVD-Laufwerken); wurde später als ATA-Standard weiterentwickelt.
IMA	Independent Management Architecture - Citrix Server-zu-Server-Infrastruktur, die stabile, sichere und skalierbare Tools zum Verwalten von Serverfarmen beliebiger Größe zur Verfügung stellt. Neben anderen Funktionen ermöglicht IMA zentralisierte, plattformunabhängige Verwaltung, einen ODBC kompatiblen Datenspeicher sowie eine Sammlung von Verwaltungsprodukten, die in die Presentation Server Console integriert werden können.
Kernel	Die Kernkomponente des Betriebssystems. Auf dem Kern bauen alle weiteren Betriebssystemkomponenten auf.
LDAP	Lightweight Directory Access Protocol bietet die Möglichkeit über TCP/IP einen Verzeichnisdienst abzufragen. LDAP ist im RFC 4511 spezifiziert. [http://tools.ietf.org/html/rfc4511]
MAC	Media Access Control, ist die Hardware-Adresse jedes einzelnen Netzwerkadapters, die zur eindeutigen Identifikation des Geräts im Netzwerk dient.
Miniinstallation	Die Miniinstallation stellt eine Untergruppe vom GUI setup dar, und stellt in der Windows-Familien den ersten Schritt dar. Sie fordert zur Eingabe benutzerspezifischer Informationen, erkennt neue Hardware und regeneriert neue Systemkennungen.
NAT	Network Address Translation (NAT) ist in Rechnernetzen der Sammelbegriff für Verfahren, um automatisiert und transparent Adressinformationen in Datenpaketen durch andere zu ersetzen.
NetBIOS	Network Basic Input Output System ist eine Programmierschnittstelle (API) zur Kommunikation zwischen zwei Programmen über ein Netzwerk. In Microsoft-Netzen werden von den 16 möglichen Zeichen 15 für Namen verwendet; das 16. Zeichen wird als Suffix benutzt, um verschiedene Dienste wie Server, RAS, Messenger usw. anzusprechen.
OSS	Open Source Software genügt i. d. R. folgenden Kriterien: 1. Die Software (d. h. der Quelltext) liegt in einer für den Menschen lesbaren und verständlichen Form vor. 2. Die Software darf beliebig kopiert, verbreitet und genutzt werden. 3. Die Software darf verändert und in der veränderten Form weitergegeben werden.
Patch	Ein Patch (von engl. Flicken, in der Bedeutung von Nachbesserung) ist eine Korrekturauslieferung für Software oder Daten aus Endanwendersicht, um z. B. Sicherheitslücken zu schließen, Fehler zu beheben oder bislang nicht vorhandene Funktionalität nachzurüsten. (s.a. Service Pack)

Plugin	Plug-in ist eine gängige Bezeichnung für ein Computerprogramm, das in ein anderes Softwareprodukt "eingeklinkt" wird. Es ergänzt dabei die Software, anders als ein Add-on stellt es jedoch eine eigenständige Software dar. Beispiel: Adobe Flash Player
PXE	Das Preboot Execution Environment (PXE) ist ein Verfahren, um Computern einen netzwerkbasierten Bootvorgang zu ermöglichen, der von client-seitig verfügbarem Massenspeicher und insbesondere Betriebssystemen unabhängig ist.
SBC	Server Based Computing, basiert auf der Technik der sechziger Jahre, wo ein Terminal an einem Mainframe angeschlossen war. In der heutigen Zeit wird anstatt des Mainframe i. d. R. leistungsstarker Server mit dem Betriebssystem Windows Terminal Server verwendet.
SCSI	Small Computer System Interface; Schnittstelle für die Verbindung und Übertragung von Daten zwischen Peripheriegeräten und Computer-Bus.
Service Pack	Service Pack (engl. für Wartungs-Paket) ist ein von verschiedenen Herstellern verwendeter Begriff für die Zusammenstellung von Patches zur Aktualisierung eines ihrer Betriebssysteme und anderer Software-Produkte.
Shell	Die Unix-Shell oder kurz Shell bezeichnet die traditionelle Benutzerschnittstelle unter Unix oder Unix-ähnlichen Betriebssystemen. Der Benutzer kann in einer Eingabezeile Kommandos eintippen, die der Computer dann sogleich ausführt.
SMB	Server Message Block ist ein Kommunikationsprotokoll für Datei-, Druck- und andere Serverdienste in Netzwerken. Es ist der Kern der Netzwerkdienste von Microsofts LAN Manager, der Windows-Produktfamilie, sowie des LAN-Servers von IBM.
SNMP	Das Simple Network Management Protocol ist ein Netzwerkprotokoll, das von der IETF entwickelt wurde, um Netzwerkelemente (z. B. Router, Server, Switches, Drucker, Computer usw.) von einer zentralen Station aus überwachen und steuern zu können.
Tarball	Zu deutsch "Teerklumpen"; mit dem Programm *tar* archivierte Dateien werden als Tarball bezeichnet.
TCO	Total Cost of Ownership bezeichnet die Summe aller finanziell relevanter Kosten, die für die Auswahl, Anschaffung, Anpassung, Installation, den laufenden Betrieb und auch die Entsorgung anfallen.
Terminalserver	Mithilfe eines Terminalservers ist es gleichzeitig mehreren Benutzern möglich, Zugriff auf den Desktop des Servers zu erhalten. Die Benutzer können von einem Remotestandort aus Programme ausführen, Dateien speichern und Netzwerkressourcen verwenden, als

	wären diese Ressourcen auf ihren eigenen Computern installiert.
TFTP	Das Trivial File Transfer Protocol (TFTP) ist ein sehr einfaches Dateiübertragungsprotokoll. TFTP unterstützt lediglich das Lesen oder Schreiben von Dateien. Es wird normalerweise über das verbindungslose Protokoll UDP betrieben und dient vorwiegend dem Laden von Betriebssystemen oder Konfigurationen über das Netzwerk.
Thin-Client	Thin-Client (engl. wörtlich "dünner Klient", inhaltlich bezogen auf Client-/Server-Architekturen) bezeichnet einen Computer als Endgerät (Terminal) eines Netzwerkes, dessen funktionale Ausstattung auf die Ein- und Ausgabe beschränkt ist. Der Thin-Client verfügt lediglich über ein stark abgespecktes Betriebssystem. Seine Daten bezieht der Thin-Client möglichst vollständig von einem Server.
TRAP	SNMP-TRAP, ist eine unaufgeforderte Nachricht von einem SNMP-Agenten an den SNMP-Manager, dass ein Ereignis eingetreten ist. Der SNMP-Manager wird durch die TRAP-Destination festgelegt.
Treiber	Betriebssystemabhängiges Stück Software, das den korrekten Zugriff und damit den Einsatz der Hardware ermöglicht.
TWAIN	TWAIN ist ein festgelegter Standard zum Austausch von Daten zwischen Bildeingabegeräten (Scanner, Digitalkameras, ...) und Programmen für Microsoft Windows und Apple Macintosh. Ein mit einer TWAIN-Schnittstelle ausgestattetes Bildverarbeitungsprogramm kann Daten von jedem Bildeingabegerät entgegennehmen, das seinerseits entsprechende Unterstützung bietet.
UUID	Universally Unique Identifier, ist ein Identifier, der bei Software und Hardware verwendet wird und weltweite Eindeutigkeit garantieren soll.
Verzeichnisdienst	Ein Verzeichnisdienst dient der Verwaltung von Informationen zu Netzwerkressourcen wie Rechnern und Druckern. Bekannte Vertreter sind Active Directory Services von Microsoft oder Network Directory Services (NDS) von Novell.
Wake On LAN	Ist ein von AMD in Zusammenarbeit mit Hewlett Packard veröffentlichter Standard, um einen ausgeschalteten Computer über die eingebaute Netzwerkkarte zu starten.